(12)

BBN Laboratories Incorporated

A Subsidiary of Bolt Beranek and Newman Inc.

AD-A187 355

DTIC FILE COPY

Report No. 6636

Research in Knowledge Representation For Natural Language Communication and Planning Assistance

Annual Report

18 March 1986 to 31 March 1987

B. Goodman, A. Haas, E. Hinrichs, H. Kautz,
L. Polanyi, J. Schmolze, M. Vilain

Prepared for:
Defense Advanced Research Projects Agency

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER 6636	2. GOVT ACCESSION NO. AD-A187355	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Research in Knowledge Representation for Natural Language Communication and Planning Assistance		5. TYPE OF REPORT & PERIOD COVERED Annual Report 3/86 - 3/87
		6. PERFORMING ORG. REPORT NUMBER 6636
7. AUTHOR(s) B. Goodman, A. Haas, E. Hinrichs, L. Polanyi, H. Kautz, J. Schmolze, M. Vilain		8. CONTRACT OR GRANT NUMBER(s) N00014-85-C-0079
9. PERFORMING ORGANIZATION NAME AND ADDRESS BBN Laboratories Inc. 10 Moulton Street Cambridge, MA 02238		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Department of the Navy Arlington, VA 22217		12. REPORT DATE October 1987
		13. NUMBER OF PAGES 124
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this document is unlimited. It may be released to the Clearinghouse, Department of Commerce, for sale to the general public.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Artificial Intelligence, Natural Language Understanding, Knowledge Representation, Semantics, Semantic Networks, KL-TWO, NIKL, Frame Problem, Reasoning, Reference, Miscommunication, Discourse, Planning, Qualitative Physics, Robotics, Temporal Reasoning

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

BBN's DARPA project in Knowledge Representation for Natural Language Communication and Planning assistance has two primary objectives: (1) To perform research on aspects of the interaction between users who are making complex decisions and systems that are assisting them with their task. In particular, this research is focused on communication and the reasoning required for performing its underlying tasks of discourse processing, planning and plan recognition and communication repair. (2) Based on the research objectives, to build tools for communication, plan recognition, and planning

DD FORM 1473, 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

assistance and for the representation of knowledge and reasoning that underlie all of these processes.

This report summarizes BBN's second year's activity in research in knowledge representation and natural language. In particular, the report discusses ~~our~~ work in the areas of knowledge representation, planning, and discourse modeling. We describe formalisms for representing knowledge necessary for the planning process. These include the representation of natural events and actions, constraint propagation algorithms for temporal reasoning, and formalisms for circumventing the frame problem. The report also contains a description of our research in discourse modelling in the area of reference. We describe how to extend the reference identification component of a natural language system to handle user's inaccurate descriptions of objects in the world and how to model the user's use of pointing gestures to refer to objects in the world. We also document publications and presentations by members of the research group over the past year.

(Keywords: artificial intelligence; semantics; robotics).

RESEARCH IN KNOWLEDGE REPRESENTATION FOR NATURAL LANGUAGE COMMUNICATION AND PLANNING ASSISTANCE

Annual Report
18 March 1986 - 31 March 1987

Principal Investigator:
Dr. Bradley A. Goodman

Prepared for:

Defense Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, VA 22209

ARPA Order No. 3414

Contract No. N00014-85-C-0079

Effective Date of Contract
18 March 1985

Contract Expiration Date
17 March 1988

Amount of Contract:
$3,476,702

Scientific Officer
Dr. Alan R. Meyrowitz

This research was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by ONR under Contract No. N00014-85-C-0079. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

This project is aimed at developing techniques to provide computer assistance to a decision maker for understanding and reacting to a complex situation. In order to provide such assistance, a system must be able to understand the needs of the decision makers through the modes of interaction in which the decision maker chooses to communicate. Central to those tasks are the abilities to understand and produce utterances with complex descriptions, to understand in the face of communication errors, and to model the discourse to connect what the speaker says to his plans and intentions and to previous requests. Handling a decision problem in a methodical way requires that planning underlies the communication task. Our research proposes new communication techniques and algorithms as well as tools for representing and manipulating knowledge. Two principal components of our research are (1) extensions of knowledge representation systems for representing time, user plans, user beliefs and miscommunication situations, and new algorithm for agents that plan to obtain new knowledge from the world and other agents, (2) natural language systems that understand the user's intentions, discourse conventions and communication errors.

The research plan we are following in this contract is aimed toward fundamental problems in Knowledge Representation and Reasoning relevant to Natural Language Communication and Planning Assistance. Central to this plan is our research in the representation of plans, plan recognition, plan formation, reasoning about plans and actions and modelling the discourse. The exploration of these research topics requires investigating both short-term as well as long-term solutions. We are also attempting to transfer some of our research to other DARPA-supported activity at BBN.

o **Natural Language Communication**. Central to extension of natural language

understanding systems from handling single (isolated) utterances to coherent dialogue is the modeling of discourse. Plans that represent the intentions of the speaker and listener are a significant component of such a model. The representation of user beliefs is an important constituent of these plans. The process of inferring the user's plans from the user's utterances (plan recognition) is the key to understanding dialogues. To further our research in understanding the intentions of utterances we ar ing the a. ty to understand in the face of miscommunication.

- **Planning Assistance**. Planning is another element underlying natural language communication. A major component of our research is the extension of knowledge representation systems for representing beliefs of agents, actions, time, continuous processes, partial hypothetical plans and multiple agents.

Our research during the past year have addressed major aspects of these problems resulting in some significant results.

- We have developed a model of discourse structure including attention and intention. This theory introduces shared plans that are developed by two agents during an extended sequence of utterances.
- We have implemented a plan recognizer.
- We have analyzed and understood miscommunication phenomena, surrounding use of noun phrase references, in actual videotaped dialogues. We are extending our theory to account for errors in recognizing the intentions underlying speaker's utterances.
- We have demonstrated a reference mechanism based on relaxation matching implemented in the KL-Two knowledge representation language.
- KL-Two has been integrated with RUP ("Reasoning Utility Package") as a reasoning utility.
- We have developed a new representation formalism that includes fluids modeled in a discrete manner based on a notion called granules and processes that are continuous are modeled discretely in a manner that permits serial as well as concurrent composition
- We have developed a logic that permits representation of nested beliefs of several agents, allows quantification with various scoping, and has efficient reasoning based on first-order unification
- As an initial exercise in parallel programming, we have a parallel unification based parser for a grammar, a grammar for natural language with excellent coverage, a running program on a Vax, Symbolics, and a parallel version on the BBN Butterfly.

An important consequence of our basic research is its transfer to on-going applications. Consistent with that view, we are transferring many of the ideas, concepts and software developed in this project to other projects both inside and outside BBN.

- o We have developed a richly expressive intensional logic language for capturing the semantics of natural language sentences, including modality, tense and context-dependence. This has now been transferred to the Strategic Computation natural language project at BBN as a meaning representation language for data base queries.
- o KL-Two has been transferred and converted from InterLisp to CommonLisp.
- o A unification based parsing algorithm and an English grammar have been transferred to the Strategic Computation natural language projects at BBN.

This report presents some of our research results in these areas over the past year. In particular, we present papers in the areas of knowledge representation for planning, semantics and discourse modelling.

Knowledge representation for planning

An area of major accomplishment this past year is in knowledge representation. We made significant progress in temporal and physical reasoning. The paper by Vilain and Kautz in this volume describes constraint propagation algorithms for temporal reasoning. In that research, they investigated computational aspects of several time representations and have shown that (1) the interval-based representation is NP-complete, (2) the point-based representation is tractable, and (3) a subset of the interval-based representation can be given a point-based representation while preserving tractability

Another paper by Schmolze details a representation formalism for modelling many natural events and actions. In that research, an architecture was designed for a robot planner in which physical knowledge of the world is distinct from and used by the knowledge of the plan/search mechanism. The physical knowledge, called Physics

for Robots (PFR), covers domains not heretofore considered actions of a robot, natural events and processes that are controlled by a robot, and objects that change over time. PFR departs from the work in Naive Physics (NP) [2, 3] in two ways. (1) PFR characterizes the robot's capabilities to act and perceive, and (2) PFR replaces the NP goal of developing models of actual common sense knowledge. Instead, PFR includes all and only the knowledge that robots need for planning, which is determined by analyzing proofs showing the effectiveness of robot I/O programs.

We also have made progress in the area of planning. The most fundamental problem in automated plan generation is the *frame problem* [4]. Briefly stated, the frame problem consists of determining those aspects of the world that are unaffected by the performance of an action. This is a considerably broader class of properties than those which are affected by the action. Most actions typically have well-circumscribed effects, and leave much of the world unchanged. Moving a box from room to room, for example, does not change its color, that of the rooms, the ambient temperature, or any other property other than the location of the box.

General solutions to the frame problem have unfortunately been very elusive. Recent efforts at formalizing planning, however, have turned to non-monotonic solutions to the frame problem. Most of these formalizations are centered around a general non-monotonic frame axiom. This axiom usually sanctions the inference that a proposition persists from some state in which it is true to a later state if it can not be proven that the proposition has been changed by an intervening action. The advantages of this approach are that it makes unnecessary the multitudinous frame axioms of the early formal systems. However the inability to prove that something has changed may be due to the incompleteness of the database, not the fact that it hasn't changed. In these cases, applying a general non-monotonic frame axiom could yield erroneous conclusions.

These problems can be avoided by a careful reconsideration of the earlier monotonic first-order planning formalisms. Indeed, it has generally been assumed that a monotonic first-order formalization of a planning domain *requires* enormous amounts of frame axioms to state all the properties that are unchanged by an action (roughly one axiom per action/property pair). In fact, all that is needed are a small number of axioms, indexed by properties of the domain. These axioms simply indicate those actions which can change a given property, all others leave it alone. These domain-specific frame axioms replace the general non-monotonic one, and sanction only monotonic inferences, thereby avoiding the problems that arose with incomplete databases in the non-monotonic formalization. This research is reported in greater depth in the paper by Haas found in this volume.

Semantics

This past year we made further progress in semantics. We developed a richly expressive logic language for capturing the semantics of natural language sentences, including modality, tense and context-dependence. That research has been transferred to the Strategic Computation natural language projects at BBN for use as a meaning representation language. One aspect of semantics addressed in that research is the locative case for prepositions involving direction such as "to" and "toward." In the paper by Hinrichs found in this report, a reformulation of the locative case for prepositions involving direction is shown. This reformulation improves upon the approach taken in case frame semantics or in conceptual dependency semantics because it is strongly compositional giving it a significant computational advantage. The semantics allows prepositional phrases involving "to" to perform in the same way with verbs of agent-changing-location such as "go" and verbs with the agent-stationary such as "wave." In previous theories each class of verbs had to be treated separately with special inference rules. In our semantics,

general principles of the representation of verbs stipulate rules that will apply for each subclass of verb.

Discourse modelling

We have demonstrated important work in discourse during the past year. The work by Sidner on a model of discourse is reported on elsewhere (see [1]). We report in this volume the work by Goodman in reference and by Hinrichs and Polanyi in pointing. Both contribute to further elucidating necessary components of a discourse model.

The paper by Goodman details how the reference component of a natural language system must be expanded to handle typical ways that users inaccurately refer to objects in the world. People often handle such poor descriptions routinely. Our goal in this work was to extend our reference mechanism to recognize and isolate such mistakes and circumvent them. In the paper we illustrate a framework less restrictive than earlier ones. We claim that relaxation is an integral part of that framework, providing a process for repairing a speaker's descriptions. Our theory incorporates the same language and physical knowledge that people use in performing reference identification to guide the relaxation process. This knowledge is represented as a set of rules and as data in a hierarchical knowledge base. Rule-based relaxation provides a methodical way to use knowledge about language and the world to find a referent. The hierarchical representation made it possible to tackle issues of imprecision and over-specification in a speaker's description. It allows one to check the position of a description in a hierarchy and to use that position to judge imprecision and over-specification and to suggest possible repairs to the description.

Pointing provides another means of referring in a discourse. In a situation where natural language and pointing facilities are combined to make an interactive

system, a unified treatment of grammar, discourse model and gesture is useful. Such a unified treatment is described in Polanyi [5]. We have demonstrated that in order to account for the contextual relevance of linguistic units such as words, phrases, sentences as well as pointings, an adequate model must include (1) a compositional syntax and semantics capable of dealing with fragmentary input and (2) a nested discourse structure that assigns a suitable interpretation context to each structure processed. The paper by Hinrichs and Polanyi describes the incorporation of referential gesture as part of their model of discourse. Pointing gestures are an important part of the communication process because they provide a concise, though vague, method of indicating to other conversational participants the intended objects of reference. Their use simplifies the language of referring expressions and provides further evidence to listeners that they have found the correct referent.

References

1. Grosz, B.J and Sidner, C.L. "Attention Intentions, and the Structure of Discourse". *Computational Linguistics 12* 3 (1986).

2. Hayes, P. The Second Naive Physics Manifesto In *Formal Theories of the Commonsense World*.
Ablex, 1985, pp. 1-36

3. Hayes, P Naive Physics 1 Ontology for Liquids. In *Formal Theories of the Commonsense World*
Ablex, 1985, pp 71-108

4. McCarthy, J., and Hayes, P. J. Some Philosophical Problems from the Standpoint of Artificial Intelligence. In *Machine intelligence 4*, B. Meltzer & D. Michie, Eds., American Elsevier, New York, 1969.

5. Polanyi, L. The Linguistic Discourse Model. Towards a Formal Theory of Discourse Structure. Technical Report 6409, BBN Laboratories Inc., Cambridge, MA, 1986.

2. CONSTRAINT PROPAGATION ALGORITHMS FOR TEMPORAL REASONING

Marc Vilain, Henry Kautz

Abstract: *This paper considers computational aspects of several temporal representation languages. It investigates an interval-based representation, and a point-based one. Computing the consequences of temporal assertions is shown to be computationally intractable in the interval-based representation, but not in the point-based one. However, a fragment of the interval language can be expressed using the point language and benefits from the tractability of the latter.*

2.1 Representing Time

The representation of time has been a recurring concern of Artificial Intelligence researchers. Many representation schemes have been proposed for temporal reasoning, of these, one of the most attractive is James Allen's algebra of temporal intervals [Allen 83]. This representation scheme is particularly appealing for its simplicity and for its ease of implementation with constraint propagation algorithms.

Reasoners based on this algebra have been put to use in several ways. For example, the planning system of Allen and Koomen [1983] relies heavily on the temporal algebra to perform reasoning about the ordering of actions. Elegant approaches such as this one may be compromised, however, by computational characteristics of the interval algebra. This paper concerns itself with these computational aspects of Allen's algebra, and of a simpler algebra of time points.

Our perspective here is primarily computation-theoretic. We approach the problem of temporal representation by asking questions of complexity and tractability. In this light, this paper examines Allen's interval algebra, and the simpler algebra of time points.

The bulk of the paper establishes some formal results about the temporal algebras. In brief these results are

- Determining consistency of statements in the interval algebra is NP-hard, as is determining all consequences of these statements. Allen's polynomial-time constraint propagation algorithm is sound but not complete for these tasks.

- In contrast, constraint propagation is sound and complete for computing consistency and consequences of assertions in the time point algebra. It operates in $O(n^3)$ time and $O(n^2)$ space.

- A restricted form of the interval algebra can be formulated in terms of the time point algebra. Constraint propagation is sound and complete for this fragment.

Throughout the paper, we consider how these formal results affect practical Artificial Intelligence programs.

2.2 The Interval Algebra

Allen's interval algebra has been described in detail in [Allen 83]. In brief, the elements of the algebra are *relations* that may exist between intervals of time. Because the algebra allows for indefiniteness in temporal relations, it admits many possible relations between intervals (2^{13} in fact). But all of these relations can be expressed as *vectors* of definite *simple relations*, of which there are only thirteen.[1] The thirteen simple relations, whose definitions appear in Figure 2-1, precisely characterize the relative starting and ending points of two temporal intervals. If the relation between two intervals is completely defined, then it can be exactly described with a simple relation. Alternatively, vectors of simple relations introduce indefiniteness in the description of how two temporal intervals relate. Vectors are interpreted as the disjunction of their constituent simple relations.

Two examples will serve to clarify these distinctions (please refer to figure 2-2). Consider the simple relations *BEFORE* and *AFTER*: they hold between two intervals that strictly follow each other, without overlapping or meeting. The two differ by the order of their arguments: today John ate his breakfast *BEFORE* he ate his lunch, and he ate his lunch *AFTER* he ate his breakfast. To illustrate relation vectors consider

[1] In fact, these thirteen simple relations can be in turn expressed in terms of universally and existentially quantified expressions involving only one truly primitive relation. For details, see [Allen & Hayes 85].

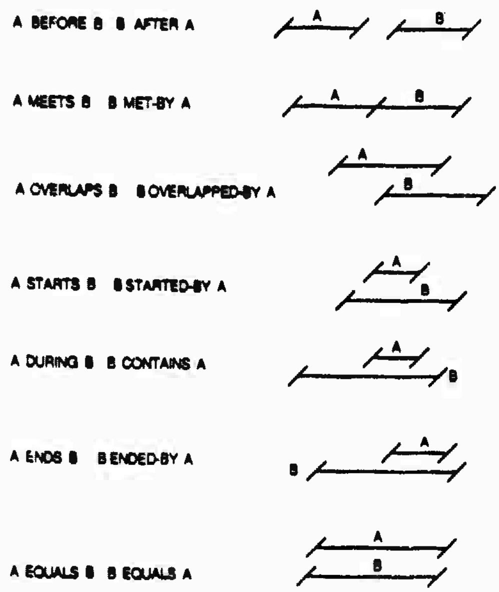


Figure 2-1: Simple relations in the interval algebra

the vector *(BEFORE MEETS OVERLAPS)*. It holds between two intervals whose starting points strictly precede each other, and whose ending points strictly precede each other. The relation between the ending point of the first interval and the starting point of the second is left ambiguous. For instance, say this morning John started reading the paper before starting breakfast, and he finished the paper before his last sip of coffee. If we didn't know whether he was done with the paper before starting his coffee, at the same time as he started it, or after, we would then have.

PAPER *(BEFORE MEETS OVERLAPS)* COFFEE

Returning to our formal discussion, we note that the interval algebra is principally defined in terms of vectors. Although simple relations are an integral part of the formalism, they figure primarily as a convenient way of notating vector relations. The mathematical operations defined over the algebra are given in terms of vectors; in a reasoner built on the temporal algebra, all user assertions are made with vectors.

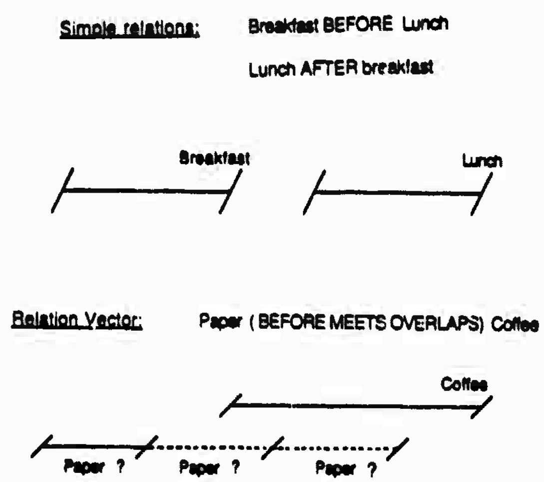


Figure 2-2: Examples of simple relations and relation vectors

Two operations, an addition and a multiplication, are defined over vectors in the interval algebra. Given two different vectors describing the relation between the same pair of intervals, the addition operation "intersects" these vectors to provide the least restrictive relation that the two vectors together admit. The need to add two vectors arises from situations where one has several independent measures of the relation of two intervals. These measures are combined by summing the relation vectors for the measures. For example, say the relation between intervals A and B has been derived by two valid measures as being both

$$V_1 = (BEFORE\ MEETS\ OVERLAPS)$$
$$V_2 = (OVERLAPS\ STARTS\ DURING)$$

To find the relation between A and B, that is implied by V_1 and V_2, the two vectors are summed.

$$V_1 + V_2 = (OVERLAPS).$$

Algorithmically, the sum of two vectors is computed by finding their common constituent simple relations.

Multiplication is defined between pairs of vectors that relate three intervals A, B,

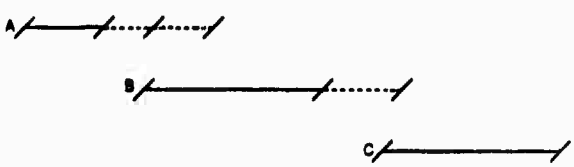
and C. More precisely, if V_1 relates intervals A and B, and V_2 relates B and C, the product of V_1 and V_2 is the least restrictive relation between A and C that is permitted by V_1 and V_2. Consider, for example, the situation in Figure 2-3. If we have

V_1 = *(BEFORE MEETS OVERLAPS)*
V_2 = *(BEFORE MEETS)*

then the product of V_1 and V_2 is

$V_1 \; x \; V_2$ = *(BEFORE)*

As with addition, the multiplication of two vectors is computed by inspecting their constituent simple relations. The constituents are pairwise multiplied by following a simplified multiplication table, and the results are combined to produce the product of the two vectors. See [Allen 83] for details.



$R<A,B>$ = (BEFORE MEETS OVERLAPS)
$R<B,C>$ = (BEFORE MEETS)

$R<A,C>$ = (BEFORE)

Figure 2-3: Intervals whose relations are to be multiplied

2.3 Determining Closure in the Interval Algebra

In actual use, Allen's interval algebra is used to reason about temporal information in a specific application. The application program encodes temporal information in terms of the algebra, and asserts this information in the database of the temporal reasoner. This reasoner's job is then to compute those temporal relations which follow from the user's assertions. We refer to this process as completing the closure of the user's assertions.

In Allen's model, closure is computed with a constraint propagation algorithm. The operation of this forward-chaining algorithm is driven by a queue. Every time the relation between two intervals A and B is changed, the pair $\langle A, B \rangle$ is placed on the queue. The algorithm, shown in Figure 2-4 operates by removing pairs from the queue. For every pair $\langle A, B \rangle$ that it removes, the algorithm determines whether the relation between A and B can be used to constrain the relation between A and other intervals in the database, or between B and these other intervals. If a new relation can be successfully constrained, then the pair of intervals that it relates is in turn placed on the queue. The process terminates when no more relations can be constrained.

As Allen suggests [Allen 83], this constraint propagation algorithm runs to completion in time polynomial with the number of intervals in the temporal database. He provides an estimate of $O(n^2)$ calls to the *Propagate* procedure. A more fine-grained analysis reveals that when the algorithm runs to completion, it will have performed $O(n^3)$ multiplications and additions of temporal relation vectors.

> **Theorem 1**: Let I be a set of n intervals about which m assertions have been added with the **Add** procedure. When invoked, the **Close** procedure will run to completion in $O(n^3)$ time.
>
> **Proof**: *(Sketch*[2]*)* A pair of intervals $\langle i, j \rangle$ is entered on *Queue* when its relation, stored in *Table*[i, j], is non-trivially updated. It is easy to show that no more than $O(n^2)$ pairs of intervals $\langle i, j \rangle$ are ever entered onto the queue. This is because there are only $O(n^2)$ relations possible between the n

---

[2]Most of the theorems in this paper have rather long proofs. For this reason, we have restricted ourselves here to providing only proof sketches.

```
/*  Let Table be a two-dimensional array, indexed by intervals, in which Table[i,j] holds the
    relation between intervals i and j. Table[i,j] is initialized to (BEFORE MEETS ... AFTER),
    the additive identity vector consisting of all thirteen simple relations; except for Table[i,i]
    which is initialized to (EQUAL). Let Queue be a FIFO data structure that will keep track of
    those pairs of intervals whose relation has been changed. Let Intervals be a list of all intervals
    about which assertions have been made. */

To   Add(R<i,j>)
    /* R<i,j> is a relation being asserted between i and j.*/

    begin
        Old ← Table[i,j];
        Table[i,j] ← Table[i,j] + R<i,j>;
        if Table[i,j] ≠ Old
            then Place <i,j> on Fifo Queue;
        Intervals ← Intervals ∪ {i, j};
    end;

To Close
    /* Computes the closure of assertions added to the database. */

    While Queue is not empty do
        begin
            Get next <i,j> from Queue;
            Propagate(i,j);
    end;

To Propagate(I,J)
    /* Called to propagate the change to the relation between
    intervals I and J to all other intervals. */

    For each interval K in Intervals do
        begin
            Temp ← Table[I,K] + (Table[I,J] x Table[J,K]);
            if Temp = 0
                then {signal contradiction};
            if Table[I,K] ≠ Temp
                then Place <I,K> on Queue;
            Table[I,K] ← Temp;
            Temp ← Table[K,J] + (Table[K,I] x Table[I,J]);
            if Temp = 0
                then {signal contradiction};
            if Table[K,J] ≠ Temp
                then Place <K,J> on Queue;
            Table[K,J] ← Temp;
        end;
```

Figure 2-4: The constraint propagation algorithm

intervals, and because each relation can only be non-trivially updated a constant number of times.

Further, every time a pair $<i,j>$ is removed from *Queue*, the algorithm performs O(n) vector additions and multiplications (in the body of the **Propagate** procedure). Hence the time complexity of the algorithm is $O(n \cdot n^2) = O(n^3)$ vector operations.

The vector operations can be considered here to take constant time. By encoding vectors as bit strings, addition can be performed with a 13-bit integer *AND* operation. For multiplication, the complexity is actually $O(|V_1| \cdot |V_2|)$, where $|V_1|$ and $|V_2|$ are the "lengths" of the two vectors to be multiplied (i.e., the number of simple constituents in each vector). Since vectors contain at most 13 simple constituents, the complexity of multiplication is bounded, and the idealization of multiplication as operating in constant time is acceptable.

Note that the polynomial time characterization of the constraint propagation algorithm of Figure 2-4 is somewhat misleading. Indeed, Allen [1983] demonstrates that the algorithm is sound, in the sense that it never infers an invalid consequence of a set of assertions. However, Allen also shows that the algorithm is incomplete: he produces an example in which the algorithm does not make all the inferences that follow from a set of assertions. He suggests that computing the closure of a set of temporal assertions might only be possible in exponential time. Regrettably, this appears to be the case. As we demonstrate in the following paragraphs, computing closure in the interval algebra is an NP-hard problem.

2.4 Intractability of the Interval Algebra

To demonstrate that computing the closure of assertions is NP-hard, we first show that determining the consistency (or satisfiability) of a set of assertions is NP-hard. We then show that the consistency and closure problems are equivalent.

Theorem 2: Determining the satisfiability of a set of assertions in the interval algebra is NP-hard.

Proof: *(Sketch)* This theorem can be proven by reducing the 3-clause satisfiability problem (or 3-SAT) to the problem of determining satisfiability of assertions in the interval algebra. To do this, we construct a

(computationally trivial) mapping between a formula in 3-SAT form and an equivalent encoding of the formula in the interval algebra.

Briefly, this is done by creating for each term *P* in the formula, and its negation *~P*, a pair of intervals, **P** and **NOTP**. These intervals are then related to a "truth determining" interval **MIDDLE**. intervals that fall before **MIDDLE** correspond to *false* terms, and those that fall after **MIDDLE** correspond to *true* terms. The original formula is then encoded into assertions in the algebra; this can be done (deterministically) in polynomial time.

The encoding proceeds clause by clause. For each clause P ∨ Q ∨ R, special intervals are created. These intervals are related to the literals' intervals **P**, **Q**, and **R** in such a way that at most two of these intervals can be before **MIDDLE** (which makes them false). The other (or others) can fall after **MIDDLE** (which makes them true).

It can then be shown that the original formula has a model just in case the interval encoding has one too. Satisfiability of a 3-SAT formula could thus be established by determining the satisfiability of the corresponding interval algebra assertions. Since the former problem is NP-complete, the latter one must be (at least) NP-hard.

The following theorem extends the NP-hard result for the problem of determining satisfiability of assertions in the interval algebra to the problem of determining closure of these assertions.

Theorem 3: The problems of determining the satisfiability of assertions in the interval algebra and determining their closure are equivalent, in that there are polynomial time-mappings between them.

Proof: *(Sketch)* First we show that determining closure follows readily from determining consistency. To do so, assume the existence of an oracle for determining the consistency of a set of assertions in the interval algebra. To determine the closure of the assertions, we run the oracle thirteen times for each of the $O(n^2)$ pairs $<i,j>$ of intervals mentioned in the assertions. Specifically, each time we run the oracle on a pair $<i,j>$, we provide the oracle with the original set of assertions and the additional assertion *i (R) j* where R is one of the thirteen simple relations. The relation vector that holds between *i* and *j* is the one containing those simple relations that the oracle didn't reject.

To show that determining consistency follows from determining closure, assume the existence of a closure algorithm. To see if a set of assertions is consistent, run the algorithm, and inspect each of the $O(n^2)$ relations between the *n* intervals mentioned in the assertions. The database is inconsistent if any of these relations is the inconsistent vector; this is the vector composed of no constituent simple relations.

The two preceding theorems demonstrate that computing the closure of assertions in the interval algebra is NP-hard. This result casts great doubts on the computational tractability of the algebra, as no NP-hard problem is known to be solvable in less than exponential time.

2.5 Consequences of Intractability

Several authors have described exponential-time algorithms that compute the closure of assertions in the interval algebra, or some subset thereof. Valdès-Pèrez [1986] proposes a heuristically pruned algorithm which is sound and complete for the full algebra. The algorithm is based on analysis of set-theoretic constructions. Malik & Binford [1983] can determine closure for a fraction of the interval algebra with the exponential *Simplex* algorithm. As we shall show below, their method is actually more powerful than need be for the fragment that they consider.

Even though the interval algebra is intractable, it isn't necessarily useless. Indeed, it is almost a truism of Artificial Intelligence that all interesting problems are computationally at least NP-hard (or worse)! There are several strategies that can be adopted to put the algebra to work in practical systems.

The first is to limit oneself to small databases, containing on the order of a dozen intervals. With a small database, the asymptotically exponential performance of a complete temporal reasoner need not be noticeably poor. This is in fact the approach taken by Malik and Binford to manage the exponential performance of their *Simplex*-based system. Unfortunately, it can be very difficult to restrict oneself to small databases, since clustering information in this way necessarily prevents all but the simplest interrelations of intervals in separate databases.

Another strategy is to stick to the polynomial-time constraint propagation closure algorithm, and accept its incompleteness. This is acceptable for applications which use a temporal database to notate the relations between events, but don't particularly require much inference from the temporal reasoner. For applications which make heavy use of temporal reasoning, however, this may not be an option.

Finally, an alternative approach is to choose a temporal representation other than the full interval algebra. This can be either a fragment of the algebra, or another representation altogether. We pursue this option below.

2.6 A Point Temporal Algebra

An alternative to reasoning about intervals of time is to reason about points of time. Indeed, an algebra of time points can be defined in much the same way as was the algebra of time intervals. As with intervals, points are related to each other through relation vectors which are composed of *simple point relations*. These primitive relations are defined in Figure 2-5.

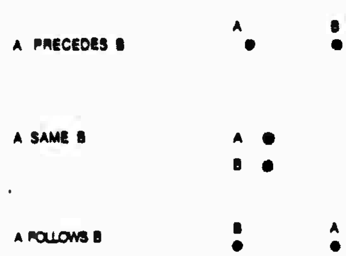


Figure 2-5: Simple point relations

As with the interval algebra, the point temporal algebra possesses addition and multiplication operations. These operations, whose tables are given in Figure 2-6, mirror the operations in the interval algebra. Addition is used to combine two different measures of the relation of two points. Multiplication is used to determine the relation between two points *A* and *B*, given the relations between each of *A* and *B* and some intermediate point *C*. These operations both have constant-time implementations if the relation vectors between time points are encoded as bit strings. With this encoding, both operations can be performed by simple lookups in two-dimensional (8 x 8) arrays. Alternatively, addition can be performed with an even simpler 3-bit logical *AND* operation.

+	<	<=	>	>=	=	~=	?
<	<	<	0	0	0	<	<
<=	<	<=	0	=	=	<	<=
>	0	0	>	>	0	>	>
>=	0	=	>	>=	=	>	>=
=	0	=	0	=	=	0	=
~=	<	<	>	>	0	~=	~=
?	<	<=	>	>=	=	~=	?

x	<	<=	>	>=	=	~=	?
<	<	<	?	?	<	?	?
<=	<	<=	?	?	<=	?	?
>	?	?	>	>	>	?	?
>=	?	?	>	>=	>=	?	?
=	<	<=	>	>=	=	~=	?
~=	?	?	?	?	~=	?	?
?	?	?	?	?	?	?	?

Key to symbols:

- 0 is () , the null vector
- < is (PRECEDES)
- <= is (PRECEDES SAME)
- > is (FOLLOWS)
- >= is (SAME FOLLOWS)
- = is (SAME)
- ~= is (PRECEDES FOLLOWS)
- ? is (PRECEDES SAME FOLLOWS)

Figure 2-6: Addition and multiplication in the time point algebra

2.7 Computing Closure in the Point Algebra

As was the case with intervals, determining the closure of assertions in the point algebra is an important operation. Fortunately, the point algebra is sufficiently simple that closure can be computed in polynomial time. To do so, we can directly adapt the constraint propagation algorithm of Figure 2-4. Simply replace the interval vector addition and multiplication operations with point additions and multiplications, and run the algorithm with point assertions instead of interval assertions.

As before, the algorithm runs to completion in $O(n^3)$ time, where n is the number of points about which assertions have been made. As with the interval algebra, the algorithm is sound: any relation that it infers between two points follows from the user's assertions. This time, however, the algorithm is complete. When it terminates, the closure of the point assertions will have been correctly computed.

We prove completeness by referring to the model theory of the time point algebra. In essence, we consider any database over which the algorithm has been run, and construct a model for any possible interpretation of the database. If the database is indefinite, a model must be constructed for each possible resolution of the indefiniteness.[3]

We choose the real numbers to model time points. A model of a database of time points is simply a mapping between those time points and some corresponding real numbers. The relations between time points are mapped to relations between real numbers in the obvious way. For example, if time point A precedes time point B in the database, then A's corresponding number is less than B's.

> **Theorem 4**: The constraint propagation algorithm is complete for the time point algebra. That is, a model can be constructed for any interpretation of the processed database.

> **Proof**: *(Sketch)* We first note that the algorithm partitions the database

---

[3]This demonstrates completeness in the following sense. If there were an interpretation of the processed database for which no model could be constructed, the algorithm would be incomplete. It would have failed to eliminate a possible interpretation prohibited by the original assertions.

into one or more partial order graphs. After the algorithm is run, each node in a graph corresponds to a *cluster* of points. These are all points related to by the vector *(SAME)*, note that the algorithm computes the transitive closure of *(SAME)* assertions. Arcs in the graph either indicate precedence (the vectors *(PRECEDES)* or *(PRECEDES SAME)*, or their inverses) or disequality (the vector *(PRECEDES FOLLOWS)*). At the bottom of each graph is one or more "bottom" nodes: nodes which are preceded by no other node.

Further, when the algorithm has run to completion the graphs are all consistent, in the following two senses. First, all points are linearly ordered, there is no path from any point in a graph back to itself that solely traverses precedence arcs (time doesn't curve back on itself). Second, no two points that are in the same cluster were asserted to be disequal with the *(PRECEDES FOLLOWS)* vector. If the user had added any assertions that contradicted these consistency criteria, the algorithm would have signalled the contradiction.

Note that all of the preceding properties can be shown with simple inductive proofs by considering the algorithm and the addition and multiplication tables.

The model construction proceeds by picking a cluster of points (i.e., a node) at the "bottom" of some graph and assigning all of its constituent points to some real number. The cluster is then removed from the graph, and the process proceeds on with another real number (greater than the first) and another cluster (either in the same graph or in another one). The process is complicated somewhat because some clusters may be "equal" to other clusters (their constituent points may be related by some vector containing the *SAME* relation). For these cases it is possible to "collapse" several (zero, one, or more) of these clusters together, and assign their constituent points to the same real number. Some other clusters may be "disequal". For these, we must just make sure never to "collapse" them together. Because the choice of which "bottom" node to remove and which clusters to collapse is non-deterministic, the model construction covers all possible interpretations of the database.

2.8 Relating the interval and point algebras

The tractability of the point algebra makes it an appealing candidate for representing time. Indeed, many problems that involve temporal sequencing can be formulated in terms of simple points of time. This approach is taken by any of the planning programs that are based on the situation calculus, the patriarch of these being *STRIPS* [Fikes & Nilsson 71]

However, as many have pointed out, time points as such are inadequate for representing many real phenomena. Single time points by themselves aren't sufficient to express natural language semantics [Allen 84], and they are very inconvenient (if not useless) for modelling many natural events and actions [Schmolze 86]. For these tasks, an interval-based time representation is necessary.

Fortunately, many interval relations can be encoded in the point algebra. This is accomplished by considering intervals as defined by their endpoints, and by encoding the relation between two intervals as relations between their endpoints. For example, the interval relation

A *(DURING)* B

can be encoded as several point assertions

A+ *(FOLLOWS)* A-
B+ *(FOLLOWS)* B-
A- *(FOLLOWS)* B-
B+ *(FOLLOWS)* A+,

where *A-* denotes the starting endpoint of interval *A*, *A+* denotes its finishing endpoint, and similarly for *B*.

This scheme captures all unambiguous relations between intervals, that is all relations that can be expressed using vectors that contain only one simple constituent. It can also capture many ambiguous relations, but not all. One can represent ambiguity as to the pairwise relation of endpoints, but one can not represent ambiguity as to the relation of whole intervals. The vector *(BEFORE MEETS OVERLAPS)* for example can be encoded as point assertions, but the vector *(BEFORE AFTER)* can not. See Figure 2-7.

The fragment of the interval algebra that can be translated to the point algebra benefits from all the computational advantages of the latter. In particular, the polynomial-time constraint propagation algorithm is sound and complete for the fragment. This is the interval representation method that Simmons uses in his geological reasoning program [Simmons 83, and personal communication].

This fragment of the interval algebra is also the one used by Malik and Binford

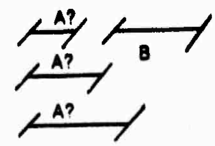
INTERVAL VECTOR	POINT TRANSLATION	ILLUSTRATION
A *(BEFORE OVERLAPS MEETS)* B	A- *(PRECEDES)* B- A- *(PRECEDES)* A+ A+ *(PRECEDES)* B+ B- *(PRECEDES)* B+	
A *(BEFORE AFTER)* B	*No equivalent point form*	

Figure 2-7: Translation of interval algebra to point algebra

[1983] in their spacio-temporal reasoning program. In their case, though, reasoning is performed with the exponential *Simplex* algorithm. This use of the general *Simplex* procedure is not strictly necessary, though, since the problem could be solved by the considerably cheaper constraint propagation algorithm.

Although many applications may be able to restrict their interval temporal reasoning to the tractable fragment of the interval algebra, some applications may not. One program that requires the full interval algebra is the planning system of Allen and Koomen [1983] that we referred to above. In this system, several actions can occur simultanesously, and must consequently be modeled with intervals. For example, to declare that two actions are non-overlapping, one asserts

ACT_1 *(BEFORE MEETS MET-BY AFTER)* ACT_2

. As we just showed, this kind of assertion falls outside of the tractable fragment of the interval algebra. In a planner with this architecture, this representation problem can be dealt with either by invoking an exponential temporal reasoner, or by bringing to bear planning-specific knowledge about the ordering of actions.

2.9 Consequences of These Results

Increasingly, the tools of knowledge representation are being put to use in practical systems. For these systems, it is often crucial that the representation components be computationally efficient. This has prompted the Artificial Intelligence community to start taking seriously the performance of AI algorithms. The present paper, by considering critically the computational characteristics of several temporal representations, follows this recent trend.

What lessons may we learn from analyses such as this? Of immediate benefit is an understanding of the computational advantages and disadvantages of different representation languages. This permits informed decisions as to how the representation components of application systems should be structured. We can better understand when to use the power of general representations, and when to set these general tools aside in favor of more application-specific reasoners.

A close scrutiny of the ongoing achievements of Artificial Intelligence enables a better understanding of the nature of AI methods. This process is crucial for the maturation of our field.

References

[Allen 83] Allen, J. F.
Maintaining Knowledge About Temporal Intervals.
Communications of the ACM 26(11):832-843, November, 1983.

[Allen 84] Allen, J. F.
Towards a General Theory of Action and Time.
Artificial Intelligence 23(2):123-154, 1984.

[Allen & Hayes 85]
Allen, J. F. and Hayes, P. J.
A Common-Sense Theory of Time.
In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, pages 528-531. The International Joint Conference on Artificial Intelligence (IJCAI), Los Angeles, CA, August, 1985.

[Allen & Koomen 83]
Allen, James F., and Koomen, Johannes A.
Planning Using a Temporal World Model.
In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, pages 741-747. The International Joint Conference on Artificial Intelligence (IJCAI), Karlsruhe, W. Germany, August, 1983.

[Fikes & Nilsson 71]
Fikes, R., and Nilsson, N.J.
STRIPS. A new approach to the application of theorem proving to problem solving.
Artificial Intelligence 2:189-208, 1971.

[Malik & Binford 83]
Malik, J. and Binford, T. O.
Reasoning in Time and Space.
In *Proceedings of the Eighth Int'l. Joint Conference on Artificial Intelligence*, pages 343-345. The International Joint Conference on Artificial Intelligence (IJCAI), Karlsruhe, W. Germany, August, 1983.

[Schmolze 86] Schmolze, J. G.
Physics for Robots: Representing Everyday Physics for Robot Planning.
PhD thesis, The University of Massachusetts, Amherst, 1986.

[Simmons 83] Simmons, R. G.
The Use of Qualitative and Quantitative Simulations.
In *Proceedings of the Third National Conference on Artificial Intelligence (AAAI-83)*. The American Association for Artificial Intelligence, Washington, D.C., August, 1983.

[Valdes-Perez 86]
Valdes-Perez, R. E.
Spatio-Temporal Reasoning and Linear Inequalities.
1986.
Unpublished A.I Memo, Massachusetts Institute of Technology Artificial Intelligence Laboratory.

3. PHYSICS FOR ROBOTS

James G. Schmolze

Abstract

Robots that plan to perform everyday tasks need knowledge of everyday physics. Physics For Robots (PFR) is a representation of part of everyday physics directed towards this need. It includes general concepts and theories, and it has been applied to tasks in cooking. PFR goes beyond most AI planning representation schemes by including natural processes that e robot can control. It also includes a theory of material composition so robots can identify and reason about physical objects that break apart, come together, mix, or go out of existence. Following on Naive Physics (NP), issues about reasoning mechanisms are temporarily postponed, allowing a focus on the characterization of knowledge. However, PFR departs from NP in two ways. (1) PFR characterizes the robot's capabilities to act and perceive, and (2) PFR replaces the NP goal of developing models of actual common sense knowledge. Instead, PFR includes all and only the knowledge that robots need for planning, which is determined by analyzing proofs showing the effectiveness of robot I/O programs.

3.1 Introduction

Physics For Robots (PFR) represents knowledge of everyday physics according to the physical capabilities and planning needs of robots. This knowledge is intended to be an important part of the overall knowledge given to a robot. Physical capabilities are represented within PFR by specifying the perceptual and action functionality of a (hypothetical) robot. This specification is comprised by an I/O programming language, whose primitive instructions correspond to primitive perceptions and actions, and an operational semantics, which describes the real world effects of executing I/O programs. (Given the complexity of the real world, this semantics is necessarily incomplete.) The hypothetical robot used for this research has capabilities that are beyond current, but are within near future technology. Some of the robot's capabilities and an I/O program are presented later in this paper.

PFR's representation of everyday physics is very similar in style to Hayes' Naive Physics (NP) formalizations [2, 3]. Like NP, PFR focuses on characterizing knowledge while postponing implementation considerations. However, NP is ultimately after realistic models of common sense (see [2], page 5) whereas PFR is after the knowledge that robots need to plan for everyday tasks. As a result, PFR includes a specification of the robot's I/O capabilities whereas NP postpones such considerations. More importantly, PFR includes a criteria for judging the value of its representations whereas NP must rely on the existing, and small, body of what is known about common sense along with one's own intuitions.

One begins to evaluate a PFR representation by selecting a set of everyday tasks for the robot to perform, and for each task, designing an I/O program that, when executed, will cause the robot to successfully perform the task. An I/O program is

one whose primitive instructions are only perceptions and actions for the robot to perform (see Section 3.4). The test for PFR is whether or not its theory of everyday physics is adequate to prove that the execution of each program will accomplish its corresponding task. The more programs/tasks that can be proven correct using a PFR representation, the greater the PFR's expressive power and the better the PFR. Further, given two expressively similar PFR representations, one should choose the simpler of the two, and one should choose the representation that is most in keeping with what is known about common sense.

I point out that there are two notions of correctness here. One is whether or not executing a program will *actually* accomplish the given task in the real world. PFR cannot be used to show this directly. For hypothetical robots, only informal arguments can used here. For actual robots, the programs can be executed and the robots observed. The second notion of correctness corresponds to whether or not executing the program accomplishes the task according to the theories of a PFR representation. The extent to which these two notions of correctness are in agreement is the extent that the representation is successful.

3.2 Composition of Materials

Physical objects in the everyday world can come into or go out of existence, break apart, come together or mix. Examples from cooking include water that boils and turns to steam, or the pouring of hot water over coffee grounds to create a cup of coffee. PFR must provide the robot with knowledge to deal with such phenomena by giving it a theory of material composition. Such a theory provides a robot with the skills to

- o identify physical objects as they come into or go out of existence, or through transformations, and
- o determine the properties of whole objects from the properties of their parts, and vice versa, including when the parts are not readily identifiable (such as the portion of the hot water that went into a cup of coffee).

My theory of material composition includes three components. (1) a theory of what constitutes the physical objects, (2) the part-whole relation along with a theory that identifies parts from wholes and vice versa, and (3) a theory that determines the properties of parts from the properties of wholes and vice versa. In this paper, I will only touch on (1) and (2), and will ignore (3) completely given that I will focus on processes. (See [5] for a fuller treatment of material composition.)

Before discussing physical objects, I now introduce some basic elements of PFR. Instants of time are represented as individuals where they form a continuum. Let "seconds" map real numbers to instants where "seconds(n)" denotes n seconds. Points in space form a 3-dimensional continuum. Changing relations are represented as functions on instants of time. Formulas and terms for these relations are written with the time argument separated. For example, "occ.space(x)(t)" denotes the set of points in space that x occupies at time t. "occ.space(x)(t)" is defined iff x is a physical object, t is an instant of time, and x exists at t. Further, x must occupy a non-empty set. Also, "vol(x)(t)" denotes the volume occupied by a physical object at time t, which is defined as the volume of "occ.space(x)(t)", and which is greater than zero for existing physical objects.

A *quantity*, borrowed from Hayes [2], is a set of measurements of a given type. For example, the temperatures and the volumes each form a quantity. Each quantity forms a continuum. I will introduce functions from the reals to various quantities, in the style of Hayes, as needed. For example, "cups(4)" denotes a volume of 4 cups.

Types that are not time-varying are called *basic* types. An example is being a physical object or a temperature. (See [5] for the reasons for the above design choices.)

Regarding notation, boolean function names, i.e., predicate names, will be capitalized

Other function names are written in all lower case. Names of constants are written in all capital letters. Names of variables are written in lower case. Variable names beginning with "t" are implicitly of type "Instant", which denotes the basic type for instants of time. I will write "$(t_1..t_2)$" to denote the open interval from t_1 to t_2. Also, I will use the following shorthand when a time varying predicate, say P, is true over an open interval.

$$[\forall t_1,t_2][P_{(t_1..t_2)} \leftrightarrow [\forall t \in (t_1..t_2)][P(t)]] \qquad (1)$$

Being a physical object is a basic type, and I write "Phys.obj(x)" when x is an individual physical object. In order to represent physical objects coming into and going out of existence, I introduce existence as a property of physical objects. Let "Exists(x)(t)" be true when x is a physical object that exists at time t. Physical objects include those objects normally considered as such, e.g., books, cars, computers, the atmosphere, oceans and glasses of water. However, for certain types of transformations that physical objects undergo, it will be useful to include very small physical objects -- possibly objects at the level of atoms and molecules. For example, the process of evaporation can be described by having small pieces of liquid turn to gas and leave the container holding the liquid. Also, by adding some sugar to water and stirring, the entire glass of water becomes sweet. By using small pieces again, one can describe mixtures and show the spread of the sweetness as a dispersion of small pieces of sugar. When hot water is poured over coffee grounds, a new object is created, coffee. It too is a mixture, which can be useful for determining that, say, the coffee is hot because it is primarily composed of pieces of water that were hot just a few seconds earlier.

Hayes ([3], page 74) eschews an atomistic theory because he considers it to be beyond the realm of common sense. In traditional physics, there is a complicated gap to bridge between the microscopic and macroscopic versions of certain properties such as temperature, volume and state. Does the robot need to know about actual atoms and molecules, and if not, what simpler theory will meet the robot's needs?

Fortunately, there is a way to meet the robot's needs without introducing microscopic versions of temperature, volume and state. To this end, I invent a class of physical objects that I call *granules*. Their essential properties are that.

- o they are small enough to be a part of all solid, liquid and gaseous physical objects -- they are too small to be seen individually,
- o they are large enough to have the usual macroscopic properties of temperature, state and volume (each has a volume greater than zero),
- o they are pure, be they purely water, wood, or whatever, and
- o they have no proper parts, and consequently, no two granules share parts nor occupied space.

Further, granules of the same type are similar. For example, two water granules with the same heat content will have the same temperature and state. Granules form the smallest physical objects in my ontology. I let "Granule" denote a basic type for granules.

By coupling the part-whole relation with granules, I have a powerful tool for describing material composition. Let "Part(x,y)(t)" be true iff x and y are physical objects that exist at t and x is a part of y at t. "Part" forms a partial order over existing physical objects at each instant. From these relations, I can define a function, called "gset", from physical objects to the sets of granules that comprise them at an instant.

$$[\forall y,t][gset(y)(t) = \{x | Granule(x) \wedge Part(x,y)(t)\}] \quad (2)$$

I will use the ability to determine an object's "gset" as the criteria for identifying the object. For example, let there be a glass called G that contains some liquid at time T. If G and T are identified to the robot, I can identify *the liquid in G* as W with the following.

$$gset(W)(T) = \{x | Granule(x) \wedge Liquid(x)(T) \wedge Contains(G,x)(T)\} \quad (3)$$

"Liquid(x)(t)" is true iff x exists and is entirely liquid at t. ("Solid" and "Gas" are defined similarly for the solid and gaseous states.) "Contains(x,y)(t)" iff x and y exist and x contains y at t. Borrowing from Hayes [3], I have used containment to identify this liquid object.

I can go a step further and write a general rule that allows the robot to identify a contained quantity of liquid as a physical object. The first line in Formula 4 requires that there is *some* liquid in a container and the remainder asserts the existence of the object formed by all the liquid in the container

$$\begin{aligned}[\forall c,t]\Big[\big([\exists x][\mathrm{Contains}(c,x)(t) \wedge \mathrm{Liquid}(x)(t)]\big) \rightarrow \quad & (4)\\ [\exists l][\mathrm{Phys.obj}(l) \wedge \mathrm{Exists}(l)(t) \wedge & \\ \mathrm{gset}(l)(t) = \{y|\mathrm{Granule}(y) \wedge \mathrm{Liquid}(y)(t) \wedge & \\ \mathrm{Contains}(c,y)(t)\}]\Big] &\end{aligned}$$

Here, x can be a single liquid granule.

Space does not permit a thorough examination of the utility of granules. The interested reader should refer to [5] where there are rules that allow the robot to identify liquid objects that are poured elsewhere, are mixed with other liquids, partially evaporate, etc. In addition, there are rules that allow the robot to infer various properties of these transformed objects, such as their temperature, volume or composition, all without special knowledge about the properties of microscopic objects. Further, the robot needs to reason about granules only when necessary; it can reason about normal physical objects without considering granules. The general PFR representation thus far allows a wide variety of such rules to be formulated. However, the actual rules for identifying objects to be given to a particular robot will be application dependent.

3.3 Simple Processes

Any robot that deals with the everyday world must be able to predict changes due to nature. An important source of natural changes is natural processes, and so, PFR includes them. I have limited my study to a class of process types that I call *simple*. All simple process types have an enabling condition and an effect, both of which depend only on the physical condition of the world (and not on, say, the intention of

any agent). Basically, an instance of a simple process type occurs when and only when the enabling condition is true for some set of physical objects, and the process has the given effect on the world while it is occurring. For example, whenever a faucet's knob is open, water flows from the faucet. Or, whenever two physical objects are of different temperatures and are in thermal contact, heat flows from the hotter to the cooler object. I note that many real processes are not simple.

Given instances of simple process types (i.e., simple processes), a robot must be able to determine when they occur, how to identify them (e.g., deciding when two processes are the same or different), and what their effects are. Further, these factors must be determinable from limited information. For example, it must be possible to determine a process' effects without knowing when the process will end. Also, the manner of describing effects must allow for either discrete or continuous changes. For example, heat is measured on a continuum, so heat transfer causes continuous changes. However, water flowing from a faucet is (eventually) measured by the transfer of whole water granules, so faucet flow causes discrete changes. Finally, the representation must allow for situations where several processes affect the same property of the very same objects, such as a heating and cooling process occurring simultaneously on the same pot of water.

I note that Hayes [2, 3] does not address these points directly. Others, such as [1] and [4] have addressed some but not all of them.

I represent simple processes as individuals. Let "Occurs(x)(t)" be true iff x is an event that is occurring at time t. "Occurs" for events is analogous to "Exists" for physical objects.

I will illustrate the essential properties of simple process types by describing the process type for water flowing from a kitchen faucet. Along with that, I will describe

faucets, objects associated with faucets (such as their controlling knobs), and their operation. Let "Faucet.flow" be a basic type for faucet flow processes. Each simple process has a set of *players*, i.e., the physical objects that are involved. For "Faucet.flow", the only player is the faucet, with which I associate other objects. In my model, a faucet has a knob, a head, a sink, a supply container that holds the faucet's supply and, of course, the water in the supply container. Let "Kitchen.faucet" and "Faucet.knob" be basic types for kitchen faucets and their controlling knobs, respectively. The knob has fully closed and fully open positions, and there are positions in between. Let "closed.position(k)(t)" denote the space that a faucet knob, k, must occupy in order to be fully closed at time t. Let "open.position(k)(t)" be similar, but for the fully open position. From these functions, I can define "Closed.knob(k)(t)" as true iff k is a faucet knob that is fully closed at t and "Open.knob(k)(t)" as true iff k is a fully open faucet knob at t.

$$[\forall k,t][\text{Closed.knob}(k)(t) \leftrightarrow \text{Faucet.knob}(k) \wedge \text{occ.space}(k)(t)=\text{closed.position}(k)(t)] \wedge \tag{5}$$
$$[\forall k,t][\text{Open.knob}(k)(t) \leftrightarrow \text{Faucet.knob}(k) \wedge \text{occ.space}(k)(t)=\text{open.position}(k)(t)]$$

If neither is true, the knob is in between. In addition, let "knob.of.faucet(f)(t)", "supply.cont.of.faucet(f)(t)" and "supply.of.faucet(f)(t)" denote the existing knob, supply container and water supply, respectively, of f when f is an existing faucet.

The enabling condition for the "Faucet.flow" process type is written over an interval of time (I will soon explain why) and is true iff a faucet, f, is not fully closed over some open interval, "$(t_1..t_2)$". The following is written with f, t_1 and t_2 free. k is used to simplify the formula.

$$[\forall t \in (t_1..t_2)][\sim\text{Closed.knob}(k)(t)] \tag{6}$$
where "k" is "knob.of.faucet(f)(t)"

I will write "Faucet.not.closed(f)(t_1,t_2)" as a shorthand for Formula 6.

The effect of a "Faucet.flow" process is that water flows from the faucet's supply container to a receiving container, which is either the faucet's sink, or an open

container under the faucet's head. To describe the effect, I rely on two defined predicates, "Liq.xfer" and "rate.liq.xfer" (only "rate.liq.xfer" will be formally presented here). "Liq.xfer(c_1,c_2,t_b,t_e)" is true iff the following holds.

1. There is some liquid in a container, c_1, at t_b.
2. Throughout the open time interval from t_b to t_e, where "$t_b < t_e$", granules from the liquid in c_1 are transferred to a different container, c_2. The transfer could have begun before t_b and could have ended after t_e. "Liq.xfer" only states that a transfer occurred throughout the particular interval "($t_b..t_e$)". Further, the liquid need not remain in c_2 (e.g., it could be transferred elsewhere).

"rate.liq.xfer(c_1,c_2,t_b,t_e)" denotes the average rate of a liquid transfer satisfying "Liq.xfer(c_1,c_2,t_b,t_e)". It is just the volume of the liquid actually transferred divided by the time of transfer. I calculate this volume by summing over the volumes of granules transferred since (1) all the liquid that is transferred may not form a single individual (e.g., if part of it was transferred elsewhere from c_2 during "($t_b..t_e$)"), and (2) granules share no parts, so I will get an accurate measurement of volume. Since the number of granules transferred is discrete, I place a minimum length on the time interval over which this rate can be calculated -- this minimum being large enough so that a reasonably large number of granules are certain to have transferred. If these intervals are allowed to be arbitrarily small, inaccurate measurements can result. Let "Δt_{Lx}" denote this minimum interval length, which I set to one-tenth second.

$$
\begin{aligned}
&[\forall r, c_1, c_2, t_b, t_e] \qquad\qquad (7)\\
&[r = \text{liq.xfer.rate}(c_1, c_2, t_b, t_e) \longleftrightarrow \text{Liq.xfer}(c_1, c_2, t_b, t_e) \wedge\\
&\quad t_e - t_b \geq \Delta t_{Lx} \wedge\\
&\quad r = \frac{1}{t_e - t_b} * \text{vol.gset}(\iota(c_1, c_2, t_b, t_e)(t_e))]
\end{aligned}
$$

where

$$
\begin{aligned}
&\iota(c_1, c_2, t_b, t_e)(t_e) = \qquad\qquad (8)\\
&\quad \{x \mid \text{Granule}(x) \wedge\\
&\qquad [\exists t_1 \in (t_b..t_e), t_2 \in (t_b..t_e)]\\
&\qquad [t_1 < t_2 \wedge \text{Liquid}(x)_{(t_1..t_2)} \wedge\\
&\qquad\ \text{Contains}(c_1, x)(t_1) \wedge \text{Contains}(c_2, x)(t_2)]\}
\end{aligned}
$$

and where "vol.gset(x)(t)" is just the volume of a set of existing granules, x, at time t.

$$\left[\forall x,y,t\right]\left[y=\mathrm{vol.gset}(x) \leftrightarrow \mathrm{Set}(x) \wedge [\forall z \in x][\mathrm{Granule}(z) \wedge \mathrm{Exists}(z)(t)] \wedge y=\sum_{z \in x}\mathrm{vol}(z)(t)\right] \tag{9}$$

"Set(x)" is true iff x is a set.

I define the effect of a "Faucet.flow" process to be that, if the faucet is fully open, water transfers from the faucet's supply container to a receiving container at the rate of one-quarter cup per second. If it is partially open, the rate is between one-sixtieth and one-quarter cup per second (this is idealized to simplify its presentation). The following describes the effect of a "Faucet.flow" process, p, that is occurring during "$(t_1..t_2)$" (remember, for p to occur, the faucet must not be closed). Let "faucet.of.flow(p)" denote the faucet involved with p. c, r and k are introduced to simply the formula.

$$\begin{aligned} &\mathrm{Liq.xfer}(c,r,t_1,t_2) \wedge \\ &\left[\mathrm{Open.knob}(k)_{(t_1..t_2)} \rightarrow \mathrm{rate.liq.xfer}(c,r,t_1,t_2)=\frac{\mathrm{cups}(1)}{\mathrm{seconds}(4)}\right] \wedge \\ &\left[\sim\left(\mathrm{Open.knob}(k)_{(t_1..t_2)}\right) \rightarrow \frac{\mathrm{cups}(1)}{\mathrm{seconds}(60)} \leq \mathrm{rate.liq.xfer}(c,r,t_1,t_2) \leq \frac{\mathrm{cups}(1)}{\mathrm{seconds}(4)}\right] \end{aligned} \tag{10}$$

where
"c" is "supply.cont.of.faucet(faucet.of.flow(p))(t)"
"r" is "receptacle.of.flow(p)(t)"
"k" is "knob.of.faucet(faucet.of.flow(p))(t)"

"receptacle.of.flow" is a function that is defined using geometrical primitives. I will not discuss it in this paper except to state that, for a "Faucet.flow" process, it refers either to the faucet's sink or to an open container directly below the faucet's head. For the formulas that follow, I will use "Effect(p)(t_1,t_2)" to refer to Formula 10.

The effect of a water flow process is written over an interval of time because there is a discrete quantity being measured, as I explained above. For this reason, I will place a minimum length on the intervals over which the effect of a faucet flow process

is calculated (as will be seen in Formula 15). Let "Δt_{eff}" denote this minimum, which, like "Δt_{Lx}", is one-tenth second. For simple process types whose effects can be measured on a continuum, "Δt_{eff}" is zero, making it possible to describe such process types using instantaneous rates, if desired. I note that enabling conditions are expressed over intervals for similar reasons, although for the enabling condition of "Faucet.flow", there is no need for a minimum length interval.

There are 5 essential properties of simple process types. For each, I include a formula written for "Faucet.flow" that describes the property. Each simple process type will have 5 similar formulas.

1. An instance begins when (or just after) the enabling condition goes from false to true for some set of players. t_b represents the beginning time for a process.

$$\begin{aligned}&\Big[\forall f{:}\mathrm{Kitchen.faucet}, t_b\Big] \qquad\qquad (11)\\&\Big[\sim[\exists t][t<t_b \wedge \mathrm{Faucet.not.closed}(f)(t,t_b)] \wedge\\&\ [\exists t][t>t_b \wedge \mathrm{Faucet.not.closed}(f)(t_b,t)] \rightarrow\\&\quad \Big[\exists p\Big]\Big[\mathrm{Faucet.flow}(p) \wedge f=\mathrm{faucet.of.flow}(p) \wedge\\&\qquad [\forall t][t<t_b \rightarrow \sim\mathrm{Occurs}(p)(t)] \wedge\\&\qquad [\forall t][t>t_b \wedge \mathrm{Faucet.not.closed}(f)(t_b,t) \rightarrow\\&\qquad\quad \mathrm{Occurs}(p)_{(t_b..t)}]\Big]\Big]\end{aligned}$$

i.e., for appropriate t_b's, a faucet flow process begins at t_b whose player -- its faucet -- is f and which continues while the faucet is not closed.

2. An instance continues as long as the enabling condition remains true for those players.

$$\begin{aligned}&\Big[\forall f{:}\mathrm{Kitchen.faucet}, t_1, t_2\Big] \qquad\qquad (12)\\&\Big[t_1<t_2 \wedge \mathrm{Faucet.not.closed}(f)(t_1,t_2) \rightarrow\\&\quad \Big[\exists p\Big]\Big[\mathrm{Faucet.flow}(p) \wedge f=\mathrm{faucet.of.flow}(p) \wedge\\&\qquad \mathrm{Occurs}(p)_{(t_1..t_2)}\Big]\Big]\end{aligned}$$

3. An instance ends when (or just before) the condition first becomes false after the process starts for those players. t_e represents the ending time for the process.

$$\left[\forall f{:}\mathrm{Kitchen.faucet}, t_e\right] \qquad (13)$$
$$\Big[[\exists t][t<t_e \wedge \mathrm{Faucet.not.closed}(f)(t,t_e)] \wedge$$
$$\sim[\exists t][t>t_e \wedge \mathrm{Faucet.not.closed}(f)(t_e,t)] \rightarrow$$
$$\Big[\exists p\Big]\Big[\mathrm{Faucet.flow}(p) \wedge f=\mathrm{faucet.of.flow}(p) \wedge$$
$$[\forall t][t>t_e \rightarrow \sim\mathrm{Occurs}(p)(t)] \wedge$$
$$[\forall t][t<t_e \wedge \mathrm{Faucet.not.closed}(f)(t,t_e) \rightarrow$$
$$\mathrm{Occurs}(p)_{(t..t_e)}]\Big]\Big]$$

i.e., for appropriate t_e's, a faucet flow process ends at t_e whose player -- its faucet -- is f and which has continued for as long as the faucet has not been closed.

4. If two individual simple processes of the same type and with the same players overlap in the times of their occurrences, they are the very same process.

$$\left[\forall p_1{:}\mathrm{Faucet.flow}, p_2{:}\mathrm{Faucet.flow}\right] \qquad (14)$$
$$\Big[\mathrm{faucet.of.flow}(p_1)=\mathrm{faucet.of.flow}(p_2) \wedge$$
$$[\exists t][\mathrm{Occurs}(p_1)(t) \wedge \mathrm{Occurs}(p_2)(t)] \rightarrow p_1=p_2\Big]$$

5. The effect applies to the players while the process occurs over intervals larger than the given minimum length.

$$\left[\forall p{:}\mathrm{Faucet.flow}, t_1, t_2\right] \qquad (15)$$
$$\left[\Delta t_{eff} \leq t_2-t_1 \wedge \mathrm{Occurs}(p)_{(t_1..t_2)} \rightarrow \mathrm{Effect}(p)(t_1,t_2)\right]$$

This knowledge allows the robot to determine when faucet flow processes begin, continue and end. It provides identity criteria for these processes and it describes their effect in the real world. Thus, the robot is well equipped to plan to control such processes. In Section 3.5, this knowledge is used to show the effectiveness of an I/O program.

3.4 Robot Perception and Action

Any robot that plans must know the consequences of executing its perceptual and action routines, i.e., its own I/O programs. In this section, I specify the I/O functionality of a hypothetical robot as part of PFR

In order to describe the effects of executing programs, a model of the robot's internal state and capabilities is needed. The robot can move about, grasp certain

kinds of objects with its (single) arm and hand, and can determine certain kinds of situations by "looking" through its (single) camera eye. Let "Near(x)(t)" be true iff the robot is near object x at t. To be near an object means that the robot is able to see it and reach it. "Grasped(x)(t)" iff the robot is grasping object x at t. In order to be grasped, the object must be of a certain shape, which I denote with "Graspable(x)(t)". Only one object can be grasped at a time. In order to represent the robot's ability to identify and find objects at given times, I introduce "Identifiable(x)(t)", which partially models the robot's internal memory state.

The I/O language includes calls to primitive input and output procedures, sequencing, compound statements, if-then-else statements and while loops. Output procedure calls are program statements. Input procedure calls are program functions. There is no assignment statement. Constants denote individuals such as physical objects or instants of time. For simplicity, I assume that the execution of the control portion of statements takes zero time. This includes calls to input procedures, so they also take zero time to execute. Also for simplicity, output procedures take fixed, greater-than-zero time to execute. In the descriptions that follow, each output procedure takes 2 seconds. (For a full specification, see [5].) Let "$E(S)(t_1,t_2)$" denote the execution of statement S by the robot where execution begins at t_1 and ends at t_2, such that a new statement can begin executing at t_2.

grasp x. If x is identifiable, graspable, near the robot and nothing is already grasped, the robot will grasp x.

$$
\begin{aligned}
&[\forall x,t_1,t_2][E(\text{grasp } x)(t_1,t_2) \rightarrow \qquad\qquad (16)\\
&\quad t_2-t_1=\text{seconds}(2) \wedge \\
&\quad (\text{Identifiable}(x)(t_1) \wedge \text{Near}(x)(t_1) \wedge \\
&\quad \text{Graspable}(x)(t_1) \wedge \sim[\exists y][\text{Grasped}(y)(t_1)] \\
&\quad \rightarrow \text{Grasped}(x)(t_2))]
\end{aligned}
$$

open.knob k. If k is a faucet knob that is currently being grasped, this causes the

robot to move k (if necessary) to its open position. It takes 2 seconds. For simplicity, I assume that the robot knows the current open position for k. If k is already open, the robot takes no action. If k is not open, it begins to move k immediately. At some point during execution of this procedure, k is in the open position, after which the robot stops moving it. Before describing "open.knob", I define "Stationary(x)(t_1,t_2)" to be true iff x does not change location from t_1 through t_2

$$[\forall x,t_1,t_2][\text{Stationary}(x)(t_1,t_2) \leftrightarrow [\forall t \in (t_1..t_2)][\text{occ.space}(x)(t) = \text{occ.space}(x)(t_1)]] \tag{17}$$

$$\begin{aligned}&\Big[\forall k{:}\text{Faucet.knob},t_1,t_2\Big]\\ &\Big[E(\text{open.knob } k)(t_1,t_2) \rightarrow t_2-t_1=\text{seconds}(2) \wedge\\ &\quad\Big(\text{Grasped}(k)(t_1) \wedge \text{Open.knob}(k)(t_1) \rightarrow\\ &\qquad \text{Open.knob}(k)_{(t_1..t_2)} \wedge \text{Stationary}(k)(t_1,t_2)\Big) \wedge\\ &\quad\Big(\text{Grasped}(k)(t_1) \wedge {\sim}\text{Open.knob}(k)(t_1) \rightarrow\\ &\qquad [\forall t \in (t_1..t_2)][\text{occ.space}(k)(t) \neq \text{occ.space}(k)(t_1)] \wedge\\ &\qquad [\exists t \in (t_1..t_2)][\text{Open.knob}(k)(t) \wedge \text{Open.knob}(k)_{(t..t_2)} \wedge \text{Stationary}(k)(t,t_2)] \wedge\\ &\qquad [\forall t \in (t_1..t_2)][\text{Open.knob}(k)(t) \rightarrow \text{Open.knob}(k)_{(t..t_2)}]\Big)\Big]\end{aligned} \tag{18}$$

close.knob k. If k is a faucet knob that is currently being grasped, this causes the robot to move k (if necessary) to its closed position. It is very similar to the "open.knob" procedure.

$$\left[\forall k{:}\mathrm{Faucet.knob}, t_1, t_2\right] \quad (19)$$
$$\Big[E(\mathrm{close.knob}\ k)(t_1,t_2) \rightarrow t_2 - t_1 = \mathrm{seconds}(2) \wedge$$
$$\Big(\mathrm{Grasped}(k)(t_1) \wedge \mathrm{Closed.knob}(k)(t_1) \rightarrow$$
$$\mathrm{Closed.knob}(k)_{(t_1..t_2)} \wedge \mathrm{Stationary}(k)(t_1,t_2)\Big) \wedge$$
$$\Big(\mathrm{Grasped}(k)(t_1) \wedge \sim\mathrm{Closed.knob}(k)(t_1) \rightarrow$$
$$[\forall t \in (t_1..t_2)][\mathrm{occ.space}(k)(t) \neq \mathrm{occ.space}(k)(t_1)] \wedge$$
$$[\exists t \in (t_1..t_2)][\mathrm{Closed.knob}(k)(t) \wedge \mathrm{Closed.knob}(k)_{(t..t_2)} \wedge \mathrm{Stationary}(k)(t,t_2)] \wedge$$
$$[\forall t \in (t_1..t_2)][\mathrm{Closed.knob}(k)(t) \rightarrow \mathrm{Closed.knob}(k)_{(t..t_2)}]\Big)\Big]$$

release: The robot releases whatever is being grasped. It takes 2 seconds.

$$[\forall t_1, t_2][E(\mathrm{release})(t_1,t_2) \rightarrow \quad (20)$$
$$t_2 - t_1 = \mathrm{seconds}(2) \wedge \sim[\exists y][\mathrm{Grasped}(y)(t_2)]]$$

Less.full(C,P): An input procedure that is true iff container C is less than a certain fraction full of solid and/or liquid material; P is the fraction. If P is 1, then this is true whenever C is not full. C must be identified beforehand and the robot must be near it. The robot estimates the value of this function using its visual capabilities along with knowledge of the container's shape. However, for this paper, this ability of the robot is idealized. Let "φ(P)(t)" be true iff the evaluation of input procedure P at time t would be true.

$$[\forall t]\Big[\mathrm{Identifiable}(C)(t) \wedge \mathrm{Near}(C)(t) \rightarrow \quad (21)$$
$$\Big(\varphi(\mathrm{Less.full}(C,P))(t) \leftrightarrow \frac{\mathrm{vol.gset}(Z)(t)}{\mathrm{contained.vol}(C)(t)} < P\Big)\Big]$$

where
"Z" is "}x|Granule(x) ∧ Contains(C,x)(t) ∧ ~Gas(x)(t){"

Here, "contained.vol(x)(t)" denotes the maximum volume of liquid material that x can contain at time t.

3.5 Filling a Pot with Water

In this section, I present an I/O program that, when executed under given conditions, will cause the robot to partially fill a pot with water. The given conditions are that a pot (P) is upright, in a sink (S), and under the head of a faucet (F) that is controlled by a knob (K) with a water supply (W) that is stored in a supply container (C). K is in the closed position. The robot is near the faucet.

FP: S_1: grasp K; (22)
S_2: open.knob K;
S_3: while Less.full(P,0.5) do idle.for seconds(0.1);
S_4: close.knob K;
S_5: release;

When FP is executed, the robot grasps K and moves K to the open position. At this point, water begins flowing into P. In S_3, the robot waits until the accumulated water occupies more than half of P. The robot then closes K and releases it, leaving P about half full of water.

PFR can be used to show the effectiveness of the FP program. The ontology and theories presented so far will be used to show that each statement of FP, when executed, produces a set of conditions needed for the next statement execution, and that at the end, the FP program has caused the robot to partially fill P with water. Furthermore, I will demonstrate how the robot has the knowledge to infer the identity of faucet flow process, even though no such process is mentioned in the FP program. I will only sketch a proof in this paper. (A full proof, excluding program termination, of a similar I/O program can be found in [5].)

I introduce T_0 through T_5, where S_1 is executed from T_0 through T_1, S_2 is executed from T_1 through T_2, etc. The relevant given conditions are

$$\begin{aligned}
&\mathrm{Faucet}(F) \wedge \mathrm{Pot}(P) \wedge && (23)\\
&K=\mathrm{knob.of.faucet}(F)(T_0) \wedge W=\mathrm{supply.of.faucet}(F)(T_0) \wedge\\
&C=\mathrm{supply.cont.of.faucet}(F)(T_0) \wedge\\
&\mathrm{Contains}(C,W)(T_0) \wedge \mathrm{vol}(W)(T_0)>\mathrm{cups}(1000) \wedge\\
&\mathrm{Exists}(F)(T_0) \wedge \mathrm{Exists}(K)(T_0) \wedge \mathrm{Exists}(P)(T_0) \wedge\\
&\mathrm{Exists}(C)(T_0) \wedge \mathrm{Exists}(W)(T_0) \wedge\\
&\mathrm{contained.vol}(P)(T_0)=\mathrm{cups}(1) \wedge \mathrm{All.water}(W)(T_0) \wedge\\
&\mathrm{Identifiable}(P)(T_0) \wedge \mathrm{Identifiable}(K)(T_0) \wedge\\
&\mathrm{Near}(P)(T_0) \wedge \mathrm{Near}(K)(T_0) \wedge \mathrm{Graspable}(K)(T_0) \wedge\\
&\mathrm{Closed.knob}(K)(T_0) \wedge \sim[\exists y][\mathrm{Grasped}(y)(T_0)]
\end{aligned}$$

Here I have used "Pot", which denotes a basic type for kitchen pots, and "All.water(x)(t)", which is true iff x is composed entirely of water granules at time t (definition not shown here).

The goal is that P contains at least half a cup of water at time T_G

$$\begin{aligned}
&[\exists l][\mathrm{Exists}(l)(T_G) \wedge \mathrm{All.water}(l)(T_G) \wedge && (24)\\
&\quad \mathrm{Contains}(P,l)(T_G) \wedge \mathrm{vol}(l)(T_G)>\mathrm{cups}(0.5)]
\end{aligned}$$

Throughout this proof sketch, I will need to make default assumptions. However, I have not investigated theories for making appropriate default assumptions in this research. Instead, I will simply make those assumptions that are needed and reasonable. As a result, I have a set of examples that a theory for making default assumptions must be able to produce. My first assumptions correspond to conditions that will not change throughout the execution of FP.

Default assumption. (25)

$$\begin{aligned}
&[\forall t \in (T_0..T_5)]\\
&[K=\mathrm{knob.of.faucet}(F)(t) \wedge W=\mathrm{supply.of.faucet}(F)(t) \wedge\\
&\ C=\mathrm{supply.cont.of.faucet}(F)(t) \wedge\\
&\ \mathrm{Contains}(C,W)(t) \wedge \mathrm{vol}(W)(t)>\mathrm{cups}(1000) \wedge\\
&\ \mathrm{Exists}(F)(t) \wedge \mathrm{Exists}(K)(t) \wedge \mathrm{Exists}(P)(t) \wedge\\
&\ \mathrm{Exists}(C)(t) \wedge \mathrm{Exists}(W)(t) \wedge\\
&\ \mathrm{contained.vol}(P)(t)=\mathrm{cups}(1) \wedge \mathrm{All.water}(W)(t) \wedge\\
&\ \mathrm{Identifiable}(P)(t) \wedge \mathrm{Identifiable}(K)(t) \wedge\\
&\ \mathrm{Near}(P)(t) \wedge \mathrm{Near}(K)(t) \wedge \mathrm{Graspable}(K)(t)]
\end{aligned}$$

Additional assumptions are needed in a complete proof, such as that certain objects do not move throughout, that the open and closed positions for K do not change, etc.

After executing S_1, the knob K is grasped. i.e., "Grasped(K)(T_1)". This follows trivially since the given condition in Formula 23 satisfies the condition of Formula 16.

While executing S_2, the robot moves K (the currently grasped object) to its open position. Let T'_1 denote the instant that K first becomes fully open, after which it remains open. T'_1 is in the interval "(T_1..T_2)". Also, according to Formula 18, the robot begins to move K immediately at T_1.

$$\text{Open.knob}(K)_{(T'_1..T_2)} \wedge \qquad (26)$$
$$[\forall t \in (T_1..T'_1)][\sim\text{Open.knob}(K)(t) \wedge \sim\text{Closed.knob}(K)(t)]$$

For similar reasons, during the execution of S_4, there is some instant when K becomes fully closed and remains closed (using Formula 19). Let this instant be T'_3, which is in the interval "(T_3..T_4)".

$$\text{Closed.knob}(K)_{(T'_3..T_4)} \wedge \qquad (27)$$
$$[\forall t \in (T_3..T'_3)][\sim\text{Open.knob}(K)(t) \wedge \sim\text{Closed.knob}(K)(t)]$$

I will now show that a "Faucet.flow" process begins at T_1 and ends at T'_3. However, first I make the default assumption that K remains fully closed during "(T_0..T_1)", fully open during "(T_2..T_3)", and fully closed during "(T_4..T_5)".

Default assumption: (28)

$$\text{Closed.knob}(K)_{(T_0..T_1)} \wedge \text{Open.knob}(K)_{(T_2..T_3)} \wedge$$
$$\text{Closed.knob}(K)_{(T_4..T_5)}$$

As a result, K is fully closed before T_1 and it is not fully closed just after T_1 (note that nothing needs to be said about K's status precisely at T_1). This satisfies the left side of Formula 11 with "t_b=T_1", leading me to conclude that there is a "Faucet.flow" process, which I'll call FF, with F as its "faucet.of.flow", that begins at T_1 and continues while K is not closed. However, Formula 11 will not let me conclude that FF ends at T'_3. Formula 13 is needed to determine process endings. Letting "t_e=T'_3" in Formula 13, I conclude that a "Faucet.flow" process, which I'll call FF2, has F as its "faucet.of.flow", ends at T'_3, and has continued for as long as K has not been closed. Of course, there is only one process here, which is concluded from Formula 14. Since

FF and FF2 use the same faucet, F, and their occurrences overlap (e.g., at T_3), then "FF2=FF".

$$\text{Faucet.flow(FF)} \wedge \text{F=faucet.of.flow(FF)} \wedge \text{Occurs(FF)}_{(T_1..T'_3)} \wedge [\forall t][t<T_1 \rightarrow \sim\text{Occurs(FF)(t)}] \wedge [\forall t][t>T'_3 \rightarrow \sim\text{Occurs(FF)(t)}] \quad (29)$$

Thus, the robot can identify a faucet flow process and can determine its times of occurrence.

Given the times of occurrence of FF, I can now determine its effect. First, I assume that P receives the water flowing from F (space does not allow a discussion of the necessary geometry).

$$[\forall t \in (T_1..T'_3)][\text{P=receptacle.of.flow(FF)(t)}] \quad (30)$$

By applying the formula describing the effects of "Faucet.flow", Formula 15, to the above times for FF's occurrence, 29, I conclude that a liquid transfer took place from C to P during "$(T_1..T'_3)$".

$$\text{Liq.xfer}(C,P,T_1,T'_3) \quad (31)$$

So, granules are accumulating in P that come from C (i.e., are part of what was W). From this, I can conclude that water is accumulating in P (and if I added more theories, that this water has properties similar to those of W, such as being either hot or cold). Also, given that FF is occurring, I can conclude the approximate rates of transfer. During "$(T'_1..T_2)$", it transfers at the maximum rate of 1 cup every four seconds. During the other times it transfers at a rate somewhere between 1 cup per minute and 1 cup per 4 seconds.

I now make the default assumptions that the liquid transferred by FF remains in P throughout execution of FP and that it remains liquid. Also, any non-gaseous object in P during execution of FP came from F's water supply, W.

Default assumption. (32)

$$[\forall x, t \in (T_0..T_5)]$$
$$[(Liquid(x)(t) \wedge Contains(P,x)(t) \rightarrow$$
$$[\forall t' \in (t..T_5)][Liquid(x)(t') \wedge Contains(P,x)(t')]) \wedge$$
$$(\sim Gas(x)(t) \wedge Contains(P,x)(t) \rightarrow$$
$$gset(x)(t) \subseteq gset(W)(T_0))]$$

Given the above, I conclude that P will continue to fill with water and that, eventually, "Less.full(P,0.5)" will be false. In fact, this will happen between 0 and 2 seconds after T_2, taking into account the varying rate of water flow and the fact that the time of T'_1 is not precisely known. Therefore, S_3 takes between 0 and 2 seconds to execute, and the entire program takes between 8 and 10 seconds. So, the robot should begin execution at "$T_0=T_G-seconds(10)$" to be sure P will be filled in time. It turns out that during the execution of S_4, another half cup of water could flow, so P will be between half and completely full.

I am nearly at the given goal, Formula 24, but it is stated in terms of a liquid object and not in terms of a set of liquid granules that are contained in P. However, Formula 4 lets the robot identify the liquid in P as a physical object, and so the goal is achieved.

3.6 Conclusions

Physics For Robots (PFR) represents the everyday physics that a robot needs to use in planning to perform everyday tasks. Using a PFR representation scheme, a robot can reason about natural processes as well as actions. It can take into account the time events take, the gradual changes they cause and the fact that many processes, once initiated, continue without further attention. Therefore, it can plan to control many processes simultaneously. PFR also specifies identity criteria for physical

objects that break apart, come together, mix, or come into or go out of existence. Therefore, the robot can plan to recognize and manipulate objects undergoing transformations, and to determine the properties of these objects based on their material composition.

The contributions of this research are:

- a strategy to develop and evaluate representations of everyday physics for robot planning,
- a general representation for part of everyday physics, including an ontology of time, space, physical objects and events, theories governing processes, material composition, etc.
- an application specific representation, describing everyday phenomena from cooking, such as water flow from a faucet, etc.

The crucial research to be done next is not only to extend these types of representations to more areas, but to use these results to design reasoning mechanisms that will allow robots to plan for everyday tasks.

3.7 Acknowledgements

Many, many thanks go to David Israel, David McDonald, Candy Sidner, Brad Goodman, N. S. Sridharan, Andy Haas, Marc Vilain and Krithi Ramamritham for their ideas and comments.

REFERENCES

[1] Forbus, K. D. The Role of Qualitative Dynamics in Naive Physics. In *Formal Theories of the Commonsense World*, Ablex, 1985, pp. 185–226.

[2] Hayes, P. The Second Naive Physics Manifesto. In *Formal Theories of the Commonsense World*, Ablex, 1985, pp. 1–36.

[3] Hayes, P. Naive Physics 1: Ontology for Liquids. In *Formal Theories of the Commonsense World*, Ablex, 1985, pp. 71–108.

[4] G.G. Hendrix. "Modeling Simultaneous Actions and Continuous Processes." *Artificial Intelligence 4* (1973), 145–180.

[5] Schmolze, J. G. *Physics For Robots*. Ph.D. Th., University of Massachusetts, February 1986. (Also BBN Laboratories Report No. 6222, July 1986)

4. THE CASE FOR DOMAIN-SPECIFIC FRAME AXIOMS

Andrew R. Haas

Abstract

Several researchers have used non-monotonic logic in attempts to abolish frame axioms that are specific to one domain in favor of a universal frame axiom We argue that the universal frame axiom cannot work in a domain that allows incomplete descriptions of situations. Therefore domain-specific frame axioms are needed. We illustrate an approach to writing these axioms by considering frame axioms about motion, including the formalization of simple example.

4.1 Introduction

The most fundamental problem in automated plan generation is the *frame problem* [7]. Briefly stated, the frame problem consists of determining those aspects of the world that are unaffected by the performance of an action. This is a considerably broader class of properties than those which are affected by the action. Most actions typically have well-circumscribed effects, and leave much of the world unchanged. Moving a box from room to room, for example, does not change its color, that of the rooms, the ambient temperature, or any other property other than the location of the box.

General solutions to the frame problem have unfortunately been very elusive. In early theorem-proving planners [4], large numbers of first-order axioms were provided to state all the properties that an action left unaffected. Not only was writing these frame axioms a painstaking endeavor, but the preponderance of these axioms in the theorem-proving database drastically curtailed the performance of the planner. As a result, planning researchers essentially abandoned first-order formalizations of actions and plans. Instead they focused on alternative solutions to the frame problem, such as the *add-* and *delete-lists* of STRIPS and its many descendants [3, 8, 1, among many others].

Recent efforts at formalizing planning have turned to non-monotonic solutions to the frame problem. Most of these formalizations are centered around a general non-monotonic frame axiom. This axiom usually sanctions the inference that a proposition persists from some state in which it is true to a later state if it can not be proven that the proposition has been changed by an intervening action. This inference is non-monotonic. It allows the planning system to *assume* something (i.e., a lack of change) on the basis of its inability to *prove* its inverse (i.e., change). The advantages of this approach are that it makes unnecessary the multitudinous frame axioms of the early formal systems. In this paper we claim that this move towards non-monotonic logic was premature. It has not been demonstrated that natural planning problems lead to huge numbers of frame axioms since only the most obvious approaches were tried. If we analyze a domain more deeply, we can reduce the number of frame axioms to something manageable without introducing any new logics.

The key is to use the standard distinction between primitive and defined predicates. For each primitive predicate there is a frame axiom which lists all the actions that can change that predicate. To prove that action A does not change a primitive predicate P, we show that A is not equal to any action on the list of actions that can change P. To show that action A does not change a defined predicate Q, we reduce Q to primitive predicates using its definition and then show that A does not change those predicates. We demonstrate by examples that in problems about motion, a non-obvious choice of primitive predicates can reduce the number of frame axioms to something quite tractable. In general, we prefer to study concrete domains rather than considering an arbitrary situation calculus.

4.2 An Argument Against the Universal Frame Axiom

Consider a problem domain in which actions are described as relations over situations and situations are described by asserting that certain conditions do or do not hold. (change A1 C1 S1) means that if the agent performs action A1 in situation S1 it will change the condition C1. There is a non-monotonic inference method whose effect is this: if the agent cannot prove the sentence (change A1 C1 S1) by some given proof procedure, then the agent concludes ~(change A1 C1 S1). We consider the case in which some situations are not completely described. This might happen either because the initial situation is not fully described, or because some of the actions are

non-deterministic. Finally, we assume there is no information about particular situations concealed in the proof procedure used in non-monotonic reasoning. Thus if the robot has limited knowledge of situation S1, the proof procedure can produce only limited predictions about the effects of performing actions in situation S1.

Given these assumptions, we claim the non-monotonic inference method cannot be reliable. Suppose the agent lacks complete knowledge of a situation S1---either because it is an underspecified initial situation or because it is the result of applying a non-deterministic action to a fully specified situation. Since the proof technique cannot make up for this ignorance, it follows that there are true statements of the form (change A2 C2 S1) that the robot cannot prove. By non-monotonic inference the robot concludes that these statements are false. Thus non-monotonic reasoning is not reliable.

One may reply that a non-monotonic inference method is allowed to make some mistakes, by definition. For this reply to be convincing one would have to show that the mistakes will be rare or unimportant. But it may well be that there are many important gaps in the agent's knowledge of the situation S1. Then there are likely to be many important statements (change A2 C2 S1) that the agent cannot prove and wrongly concludes are false. The universal frame axiom is reliable only when nearly all important facts about each situation are available.

We have assumed that the non-monotonic inference method applies to all situations, no matter how little the agent knows about them. One alternative is a non-monotonic inference method that applies only when the agent knows the relevant facts about the starting situation. We might have "If the agent knows the location of every object that is inside room R in situation S, and action A1 takes place inside room R, and the agent cannot prove (change A1 C S1), then ~(change A1 C S1)". Such an axiom may indeed be useful, but it is not domain-independent - that is, it does not apply to every domain in which actions are relations over situations. Yet this is what advocates of the universal frame axiom mean by "domain-independent". This suggests that a weaker notion of domain-independence might be useful, and we will return to this possibility.

Hanks and McDermott [5] pointed out that even in the case of a deterministic domain, it is not obvious how to formalize a domain-independent frame axiom like "if you can't prove action A changes condition C in situation S, assume it doesn't".

Various authors (e.g., Kautz [6]) have shown that one can do it by applying this frame axiom to the steps of a plan in order of their occurence These solutions are all domain-independent, so by the above argument they ought to produce false conclusions in cases where there are incomplete descriptions of situations. For simplicity assume the initial state is underspecified. Suppose the axioms say that the action Shoot changes the condition Alive in any situation S where Alive and Loaded are true. Suppose the initial situation S0 is not completely described---it is known that Loaded is true in S0, but not known whether Alive is true in S0 or not. Then the agent cannot prove that Shoot changes Alive in S0, and it concludes that Shoot does not change Alive in S0. It then follows by ordinary monotonic logic that Alive was false in the initial situation. This is the wrong conclusion---at least according to my intuition about what the frame axiom was supposed to mean.

If the initial situation is fully specified, and all the actions are deterministic, then applying the frame axiom to the actions in order of their occurence will guarantee that each situation is fully described when the frame axiom is applied to it. Only then will the universal frame axiom produce correct results. Suppose instead that we use frame axioms which explicitly require that the relevant knowledge is available - such as the room axiom above. If relevant facts about a situation are missing because the frame axioms have not yet been applied to the step that produced that situation, then these frame axioms will not apply to that situation. They will also not apply if the relevant facts are missing for some other reason---for example, because of a non-deterministic action. Thus by using frame axioms which explicitly require the relevant knowledge, we can drop the constraints that the initial situation is fully specified, all actions are deterministic, and the frame axiom is applied to the steps of a plan in order of occurence. These are really only consequences of a general principle. default rules cannot be applied safely unless all the relevant knowledge is available.

4.3 Domain-Specific Frame Axioms

People usually discuss the frame problem quite abstractly, with little attention to concrete plans. This has gone so far that Hanks and McDermott created a stir by discussing a concrete plan of three steps, in a domain with three predicates! Davis [2] looked at a less trivial domain---a subset of classical mechanics. He found that no

frame axioms at all were needed. He had to assume, in each example, that a short list of objects were the only ones involved in the example. Given that, the general axioms about force and motion sufficed to prove that objects remained in their places. The present paper will not offer a general discussion of domain-specific frame axioms, instead we consider one particular problem involving frame axioms about motion, and we illustrate by formalizing a simple example. This will illustrate our approach to writing domain-specific frame axioms.

If you load your possessions in the trunk of your car and drive it across town all your possessions will move across town, but they will remain in your car. How can we describe this situation with reasonable frame axioms? We rely on the distinction between primitive and defined predicates, used in STRIPS. There are frame axioms only for primitive prediates, to prove the persistence of defined predicates we reduce them to primitive predicates using their definitions. (at object spot) says that an object is resting on the street at a certain spot, (in object container) says that an object is sealed in a container. These are primitive predicates. (location object spot) is a defined predicate, it holds if either (at object spot) or there is an object2 such that (at object2 spot) and (in object object2). A frame axiom says that driving will not delete (in object car). Then if driving across town makes (at car spot1) true while preserving (in object car), we can deduce that (location object spot2) holds.

In general, an assembly is a collection of objects that can be moved as a group. My car and its contents form an assembly, if I am holding an object then I and that object belong to an assembly. The planet Earth and all its contents form a very large assembly. Primitive predicates describe the location of an object only with respect to the smallest assembly that contains that object. To describe an object's location with respect to a larger assembly, we use defined predicates. Frame axioms assert that in certain cases we can move an assembly while preserving the locations of its parts with the respect to the assembly. Given the new location of the assembly in some large scale frame of reference, we can infer the new locations of its parts in that large scale frame of reference.

Consider a domain based on the STRIPS domain of rooms and boxes. The domain contains five kinds of objects: boxes, rooms, doors, spots, and the robot itself. Spots are pieces of floor inside the rooms, and an whenever an object is resting on the floor it is in exactly one spot. The robot is always resting on the floor, but the boxes are not. A box may rest on the floor, or the robot may be holding it, but not both.

The robot can hold many objects at once. We use the following predicates, with variables marked by colons.

(result :s :a :s')	Action :a turns situation :s into :s'.
(at :obj :sp :s)	Object :obj is at spot :sp in situation :s.
(next :sp1 :sp2)	The spot :sp1 is next to the spot :sp2.
(holding :b :s)	The robot is holding box :b in situation :s.
(in :sp :r)	Spot :sp is in room :r.
(connect :d :r1 :r2)	Door :d connects room :r1 and room :r2.

Only two predicates have situation arguments, because only these predicates can be changed by actions. The robot and the boxes it is holding form an assembly, the primitive predicate "holding" describes the location of a box with respect to that assembly.

The actions are as follows.

(pickup :b)	The robot picks up box :b.
(putdown :b)	The robot puts box :b down.
(goto :b)	The robot goes to the location of object :b (which must be in the same room as the robot).
(gothru :d)	The robot goes through door :d.

We assume that if f and g are distinct terms and denote actions, they denote distinct actions. This assumption can be implemented with an axiom schema.

Each action has an axiom to describe the changes that it causes when executed with the correct preconditions. The robot can pick up a box if the robot is at a spot next to the box's spot.

[(at :b :sp1 :s) & (at Robot :sp2 :s) & (next :sp1 :sp2)
& (result :s (pickup :b) :s')] $\rightarrow$ (holding :b :s')

It is not neccesary to postulate that this action deletes the old location of :b, that follows because an object cannot be in a spot when the robot is holding it.

The robot can put down any box that it is holding

[(at Robot .sp .s) & (holding :b .s) & (result .s (putdown :b) :s')]
→ (∃ .sp'. (at .b .sp' .s') & (next .sp .sp'))

The robot can go to a spot next to box .b if .b is in the same room as the robot.

[(at .b .sp1 .s) & (at Robot .sp2 .s) & (in .sp1 .room) & (in .sp2 .room)
& (result :s (goto :b) .s')]
→ (∃ .sp3 . (at Robot .sp3 .s') & (next .sp3 sp1))

The robot can go from room to room through a door.

[(at Robot .sp1) & (in .sp1 :room1) & (connect .door :room1 :room2)
& (result :s (gothru .door) :s')]
→ (∃ :sp2 . (at Robot .sp2 :s') & (in :sp2 .room2))

For each primitive predicate there are two classes of frame axioms. those that limit the class of actions that add the predicate, and those that limit the class of actions that delete the predicate. For the predicate "at", we consider first the movement of the robot. Only GoTo and GoThru can add or delete "at" for the robot.

[(at Robot :sp :s) & ~(at Robot :sp .s') & (result .s .act .s')]
→ [(∃ :b . :act = (GoTo .b)) v (∃ .d . .act = (GoThru .d))]

[~(at Robot :sp :s) & (at Robot :sp :s') & (result .s .act .s')]
→ [(∃ .b . :act = (GoTo :b)) v (∃ .d . .act = (GoThru .d))]

For boxes, only Pickup deletes "at", and only Putdown adds "at".

[(IsBox .b) & (at :b .sp .s) & ~(at .b .sp .s') & (result .s .act .s')]
→ .act = (pickup .b)

[(IsBox .b) & ~(at .b .sp .s) & (at .b .sp .s') & (result .s .act .s')]
→ .act = (putdown .b)

Putdown is the only action that deletes "holding", and Pickup is the only action that adds "holding".

[(holding .b .s) & ~(holding .b .s') & (result .s .act .s')]
→ .act = (putdown .b)

[~(holding :b .s) & (holding :b .s') & (result .s .act .s')]
→ .act = (pickup .b)

These axioms allow us to prove, for example, that when the robot travels from one room to another the set of boxes it is holding does not change. Suppose (InRoom .b .r) holds when box .b is in room .r. This predicate cannot be primitive because it describes the location of a box with respect to the large scale frame of reference, even when that box is being held by the robot and so forms part of an assembly. By contrast, the primitive predicate "at" describes a box's location with respect to a large scale frame of reference, but only when the box is not part of an assembly – that is, the robot is not holding the box. The predicate "InRoom" can of course be defined – (InRoom .b :r :s) holds iff

(∃ :sp . (in :sp .r) &
[(at .b :sp :s) v [(at Robot .sp .s) & (holding .b :s)]]
)

One can then prove that if the robot goes to room A while holding block B1, the robot will still be holding B1 and so B1 will be in room A.

The notion of an assembly is certainly not domain-independent in the sense that it applies to any domain where actions are relations over situations. Yet it does apply to a rich variety of situations in everyday life---from carrying a cup of coffee across the room to flying a plane across country. We argue that this kind of "domain independence" is more feasible and more useful than the kind of domain independence that the universal frame axiom was supposed to have. There may be many ideas which, like the notion of an assembly, are useful in a wide variety of everyday situations. Planning research should aim to discover ideas like these, rather than ideas that apply to any domain containing situations and actions. People who build theorem provers are now less interested in universal first-order theorem provers and more concerned with programs designed to reason about crucial concepts like time or belief. People who study the frame problem should likewise look not for a universal solution, but for ideas that apply often in the world of common sense and everyday life.

References

1. Chapman, David. Nonlinear Planning. A Rigorous Reconstruction. Proceedings of the Ninth International Joint Conference on Artificial Intelligence, IJCAI, Los Angeles, California, August, 1985, pp. 1022-1024

2. Davis, Ernest. A Logical Framework for Solid Object Physics. 245, Department of Computer Science, New York University, October, 1986.

3. Fikes, R., and Nilsson, N.J. "STRIPS. A new approach to the application of theorem proving to problem solving". *Artificial Intelligence 2* (1971), 189-208.

4. Green, C. Application of theorem-proving techniques to problem-solving. Proceedings of the international joint conference on artificial intelligence, Washington, D. C., May, 1969.

5. Hanks, Steve and McDermott, Drew. Default Reasoning, Non-Monotonic Reasoning, and the Frame Problem. Procedings of AAAI-86, American Association for Artificial Intelligence, 1986, pp. 328-333.

6. Kautz, H.A. The Logic of Persistence. Proceedings of AAAI-86, the 5th National Conference on Artificial Intelligence, American Association of Artificial Intelligence, August, 1986, pp. 401-405.

7. McCarthy, J., and Hayes, P. J. Some Philosophical Problems from the Standpoint of Artificial Intelligence. In *Machine intelligence 4*, B. Meltzer & D. Michie, Eds., American Elsevier, New York, 1969.

8. Sacerdoti, E. D.. *A Structure for Plans and Behavior*. American Elsevier, New York, 1977.

5. A COMPOSITIONAL SEMANTICS FOR DIRECTIONAL MODIFIERS

Erhard W. Hinrichs

Abstract

This paper presents a model-theoretic semantics for directional modifiers in English. The semantic theory presupposed for the analysis is that of Montague Grammar (cf. Montague 1970, 1973) which makes it possible to develop a strongly compositional treatment of directional modifiers. Such a treatment has significant computational advantages over case-based treatments of directional modifiers that are advocated in the AI literature.

5.1 Case-based Treatments

Among natural language processing systems which attempt to incorporate spatial information, the following strategy seems to prevail. Directional or locative modifiers are treated either as corresponding slots in *case frames* in the canonical lexical representations of verbs (cf. Celce 1972, Hendrix, Thompson and Slocum 1973), or as corresponding to *conceptual cases* in the (meta-linguistic) conceptualization of actions (Schank 1975).

Case based approaches to the semantics of directional modifiers can be characterized as *weakly compositional* in the following sense. In a verb phrase such as *fly to Chicago* the prepositional phrase contributes semantically the meaning of the NP *Chicago* as the directional or locative goal of the action associated with the verb phrase. However, the directional preposition *to* itself does not make a semantic contribution at all to the meaning of the verb phrase as a whole. Instead, *to* merely serves as a syntactic marker for a semantic entity, namely locative or directional case whose meaning cannot be separated from, but rather is an integral part of a given verb frame or conceptual structure. By contrast, the semantics of directional modifiers that I will be advocating in this paper is *strongly compositional* in the sense that directional prepositions serve as autonomous syntactic and semantic units.

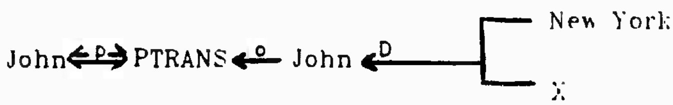
Consequently, each word in a phrase such as *fly to Chicago* contributes its own, independent meaning to the meaning of the phrase as a whole.

This strongly compositional analysis of directional modifiers has a number of crucial computational advantages over case-based approaches. Consider how inferences between sentences such as (1) and (2) can be handled by the two types of approaches.

(1) John went to New York.
(2) John was in New York.

In Schank (1975, p.53) sentence (1) corresponds to the conceptual structure in (3).

(3)



(3) be should read as John is at some time in the past (p) engaged in an act of physical transfer (PTRANS) whose object (o) is John and whose direction (D) is from some location X to New York. The fact that (1) implies (2) is expressed by attaching to the bi-directional arrow in (3) the structure in (4). (cf. Schank 1975, p. 54)

(4)

Schank calls the *r-link* (*r* for *result*) between structures (3) and (4) an *inference*. However, the term *inference* is really a misnomer because the association between structures such as (3) and (4) is merely a matter of stipulation but does not follow from any general principles or axioms that would constrain the language of conceptual structures. For that matter, there is nothing in Schank's system that prevents a link between (3) and a structure which expresses that John does not reach the location *New York*. In the analysis we will develop below, on the other hand, the inference between (1) and (2) follows logically from the semantics of motion verbs such as *go* in conjunction with the semantics of directional modifiers.

Consider next the issue of how easy or difficult it is to upscale natural language systems whose treatment of directional modifiers is case-based. Assume a case-based system in which only those verbal frames or conceptual structures are implemented that relate locative or directional case to verbs of motion. Now imagine that we want to extend coverage to verbs such as *wave* which, as illustrated in (5), allow directional modifiers such as *to*.

(5) The President waved to the reporters.

Since *wave*, unlike verbs of motion, does not entail a change of location for the agent involved, a new verbal frame or conceptual structure would have to be introduced into a system which only covers motion verbs. Moreover, locative or directional case would have to be reintroduced into the system as well because in a case-based system the specific effect of a given semantic case has to be determined for each individual frame or conceptual structure. This is a direct consequence of the weakly compositional semantics of such systems and in turn leads to an highly redundant method of upscaling. Since our analysis of directional modifiers is, by contrast, strongly compositional, upscaling becomes much easier. In the case of extending coverage to a verb like *wave*, all that needs to be added is the lexical semantics for the verb itself, while the semantics of directional modifiers can remain untouched.

Finally, consider how a case-based approach to directional modifiers fares with respect to phrases such as the ones given in (6).

(6) From Russia with Love
To New York and then to Atlanta

Since in case-based systems locative or directional case is a relational notion and is crucially dependent on a verbal frame or conceptual structure, it becomes impossible to assign an interpretation to verbless phrases as in (6). One strategy for extending case-based systems to such verbless phrases would consist in supplementing the relational notion of directional or locative case by a non-relational counterpart which does not depend on some verbal frame or conceptual structure. But the resulting account of locative or directional case would once again be highly redundant since essentially all of the cases in the system would have to be split into a relational and a non-relational version.

5.2 Motion Verbs as Location Predicates

In their literal sense, locative use *to* and *toward* typically modify motion verbs such as *walk*, *run*, *drive*, *slither*, *move* etc. An adequate treatment of the directional modifiers themselves is, therefore, closely connected to a semantic account of such motion verbs. The treatment of motion verbs that I will adopt in this paper is that developed in Hinrichs (1985) where I argue that motion verbs should be treated as

stage level predicates in the sense of Carlson (1977), namely as predicates whose arguments refer to stages of individuals. Stages are connected to individuals in Carlson's ontology by a realization relation R, which associates a given individual with all of the (spatio-temporal) stages at which that individual is present.

Motion verbs such as *move* can be understood as prototypical examples of stage-level predicates, since such verbs predicate something about the spatio-temporal location of one or more objects. Following Hinrichs (1985), I interpret a motion verb like *move* in terms of a three-place stage level predicate $move^{+}$, whose first two argument positions range over individual stages realizing the referents of the object and subject NPs, respectively. The rightmost argument position ranges over event stages realizing the event that the referents of the subject and object NPs are engaged in. Thus, $move^{+}(x^{S})(y^{S})(e^{S})$ should be read as "the referents of x^{S} and y^{S} are engaged in an event stage e^{S} realizing an event of moving.". As is customary in Montague Grammar, we express constraints on lexical meaning in terms of meaning postulates that constrain the set of possible models of semantic interpretation.[1] The meaning postulate in (7) states that an event stage e^{S} which realizes a moving event spatio-temporally includes (symbolized as $\leq$) at least the location of the referent denoted by the object argument, i.e. $y^{s} \leq e^{s}$ This does not exclude the possibility that the location of the referent of the subject NP can be contained in the event stage as well, but this is not required for *move* in view of examples like (8).

(7) $\forall\ x^{s},y^{s},e^{s}\ [\ move^{+}(x^{s})(y^{s})(e^{s}) \longrightarrow y^{s} \leq e^{s}]$
(8) John moved the troops.

Of course, different motion verbs will have different properties with respect to the relative locations of event stages and those stages that realize the individuals involved in these event stages. Take verbs like *slither*, *walk*, and *run*, for example which in our framework are analyzed as two-place stage level predicates. For these predicates it seems reasonable to simply equate the location of the event stage with the location of the agent, i.e. the referent of the subject NP. This can be enforced by a meaning postulate as in (9)

(9) $\forall\ x^{s},e^{s}\ [\delta^{+}(x^{s})(e^{s}) \longrightarrow x^{s}=e^{s}\ ]$, where δ translates *slither*, *walk*, *run*, *etc.*.

---

[1] All the meaning postulates appearing in this paper are formulated in the language of extensional logic developed in Hinrichs (1985).

The lexical entailment associated with the verb *move* to the effect that the location of the referent of the object NP changes can be captured by the meaning postulate in (10). (The symbols $<$ and $\neq_s$ used in (10) stand for temporal precedence and spatial inequality, respectively.)

(10) $\forall\, e^s, x_1^s, y^s, x^o\ [R(x_1^s, x^o)\ \&\ move^+(x_1^s)(y_1^s)(e_1^s) \longrightarrow \exists\, x_2^s\ [R(x_2^s, x^o)\ \&\ x_1^s < x_2^s\ \&$
$x_1^s \neq_s x_2^s]$

5.3 The Semantics of *to* and *toward*

Now we let a *to*-phrase, as a modifier of untensed verb phrases (IV*), operate semantically on the event stages in the denotation of the unmodified verb phrase in such way that the event stages in the denotation of the resulting IV* phrase constitute a spatio-temporal path (in the sense of Cresswell 1978) between some specified point of origin to the location of the term combining with *to*. The translation of *to* is given in (11).

(11) *to* translates as $\lambda P \lambda P \lambda l_1 \lambda x^i P\ [\lambda y^i \exists l_2 [R(l_2, y^i)\ \&\ PATH(l_1, l_r, l_2)\ \&\ P(x^1)(l_1)]]$

Ignoring for the time being the battery of lambda-abstractions at the beginning of the formula, which make the translation of *to* come out to be of the right semantic type, the formula following the lambda abstractions introduces an individual stage l_2 realizing an individual object y^i, which is the one bound by the noun phrase (NP) combining with *to* to form the IV* modifier. The second conjunct in the formula asserts that the denotation of the event stage located at l_1, which is to be bound by the translation of the IV* phrase that the *to*-phrase combines with, qualifies as a *spatio-temporal path* (a notion formally defined in Hinrichs 1985) between some point of origin l_r and the spatio-temporal location of the point of destination. Finally, the third conjunct asserts the truth of the unmodified IV* phrase that the the *to*-phrase combines with. It is this last conjunct that automatically guarantees the inference from sentences such as (12) to sentences such as (13).

(12) Fangs slithered to the rock.
(13) Fangs slithered.

Using the translation for *to* suggested in (11), sentence (12) receives the reduced translation in (14) according to our analysis.
Paraphrasing (14), it says that there is an event stage realizing some individual event

(14) $\exists e^s, e^i\ [R(e^s, e^i)\ \&\ PAST(e^s)\ \&\ \exists x^s\ [R(x^s, f)\ \&\ \exists x^o \forall z^o\ [\ rock'(z^o)\ \&\ \exists z^s$
$[R(z^s, z^o)\ \&\ slither'(x^s)(e^s)\ \&\ PATH(e^s, l_r, z^s)] \longleftrightarrow x^o = z^o]]$

of Fangs' slithering such that that event stage lies in the past and the spatio-temporal location of the event stage constitutes a path between some implicit point of reference l_r and the location of some unique rock object The point of reference l_r occurs as a free variable in the formula in (14). l_r is to be understood as an indexical parameter similar to the notion of a *reference point* proposed by Reichenbach (1947) for the interpretation of tenses in English.

Notice that the notion of a path in the translation of *to* in (11) and hence also in the translation for (12) given in (14) is defined to hold of the process making up a particular event. Moreover, due to the postulate in (9), the referent of the subject NP, when it combines with a motion verb such as *slither to the rock*, is realized by a stage spatio-temporally co-extensive to the path denoted by the *to* phrase. This fact guarantees the inference between sentences such as (12) and (15).

(15) Fangs was at the rock.

For other classes of verbs the same type of inference, namely identifying the path with the position(s) of the referent of the subject NP, cannot be drawn. For sentences such as (16) we do not want to claim that the stages realizing John make up a path to Boston. Rather it is the object NP, in this case an event term, that constitutes the path. The same is true of (17), it is the ball whose locations constitute a path to the location specified in the *to*-phrase.

(16) John made a phone call to Boston.
(17) Carol set the ball to Lucy.

Let us now turn to the treatment of the preposition *toward* whose lexical translation rule is given in (18)

(18) *toward* translates as $\lambda \mathcal{P} \lambda P \lambda e^s \lambda x^i \mathcal{P}(\lambda y^i\ \exists l\ [R(l, y^i)\ \&\ \exists l'\ [PATH(l', l_r, l))\ \&\ e^s \cdot$
$l'\ \&\ l_r \cdot e^s\ \&\ P(x^i)(e^s)]])$

The translation for *toward* constrains the value of the event stage variable e^S (to be bound by the stage-level predicate of the IV* with which the *toward*-phrase combines) The value for e^S has to be spatio-temporally contained in some initial segment of a path l' from some implicit point of origin l_r to the location l of the referent of the NP with which *toward* The requirement that the value of e^S has to be

an initial segment of such a path follows from the condition that the implicit point of origin l_r has to be properly contained in e^S. *Proper* containment is necessary in order to avoid that the value of e^S could be equal to the point of origin, in which case an object could count as moving toward another object if the spatial location of the first object remains unchanged.

Let us demonstrate how the rule in (18) applies in the translation of sentence (19) that is given in (20)

(19) Fangs slithered toward the rock.

(20) $\exists e^s, e^i\, [R(e^s, e^i)\ \&\ PAST(e^s)\ \&\ \exists x^s\, [R(x^s, f)\ \&\ \exists x^o\, [\forall z^o\, [rock'(z^o) \longleftrightarrow x^o = z^o]$
$\&\ \exists z^s\, [R(z^s, z^o)\ \&\ slither'(x^s, e^s)\ \&\ \exists l\, [PATH(l, l_r, z^s)\ \&\ e^s \leq l\ \&\ l_r < e^s]]]]]$

The translation in (20) says that there is an event stage realizing some individual event of Fangs' slithering such that that event stage lies in the past and the spatio-temporal location of the event stage constitutes the initial part of a path between some implicit point of reference l_r and the location of some unique rock object. Since e^S in (20) is an initial part of a complete path to the rock, the truth of a sentence such as (12) entails the truth of (19), but not vice versa. Moreover, (12), but not (19), entails (15).

5.4 The Aspectual Effect of *to* and *toward*

Apart from accounting for the relevant inference patterns between sentences such as (12), (15) and (19), an adequate analysis of *to* and *toward* should also for a systematic difference in the aspectual behavior of these two directional modifiers. Sentences such as (21a) which involve the preposition *to* describe *atelic events* or, in the terminology of Vendler (1967), *activities*. Sentences such as (21b), on the other hand, refer to telic events or to *accomplishments* in the Vendler classification.

(21) a. John walked to the library.
b. John walked toward the library.

These aspectual properties can be demonstrated by examining the cooccurrence restrictions of the sentences in (21) with temporal modifiers such as *in an hour* as in (22) and with *for an hour* as in (23)

(22) a. John walked to the library in an hour
b. * John walked) toward the library in an hour

(23) a. John walked to the library for an hour.
b. John walked toward the library for an hour.

As first pointed out by Vendler, only telic events or accomplishments can occur with temporal modifiers such as *in an hour*. Modifiers such as *for an hour* can occur with both activities and accomplishments. However, when modified by temporal *for*, only activities as in (23a) can be interpreted as describing a single event. If temporal *for* occurs with sentences that describe accomplishments as in (23b), such sentences have to be interpreted in some special fashion to make them semantically acceptable. (23b), for example, can best be understood as referring to an iterative event, namely of John's repeatedly walking to the library during the period of one hour.

Since doing something *for* x amount of time means doing something during most if not all subintervals of the interval x, sentences such as (24), which refer to atelic events or activities, can be characterized as being temporally homogeneous.

(24) Fangs slithered toward the rock.

To do something *in* x amount of time, on the other hand, means to do something at some unique interval within x. Since telic events or accomplishments can be modified by temporal *in*, they, in contrast to activities or atelic events, can be described as being temporally heterogeneous: telic events such as (25) come about over the course of some unique time interval I', i.e. not at some subinterval of I' or at some interval properly containing I'.

(25) Fangs slithered to the rock.

If my analysis of directional *toward* and *to* is an adequate one, it should predict that verb phrases formed with directional *toward* refer to temporally homogeneous events, while verb phrases formed with *to* refer to temporally heterogenous events. Due to the way in which I have defined *toward* as an initial subpart of a path to the projected point of destination, the reference property of temporal homogeneity associated with *toward* can, in fact, be reconstructed in the following way. Let us assume that there is a location l_1 which qualifies as an initial segment of a path from a putative point of origin r_1 to a destination d. Moreover, let us assume that r_2, the temporally final bound of l_1, is in turn the temporally initial bound for a location l_2 which forms the intial part of a path from r_2 to d. Then it follows that $l_1 \oplus l_2$ is an initial segment of a path from r_1 to d. This is precisely what is required to make the semantics of *toward* homogeneous.

Since our account of motion verbs and directional *toward* does predict that sentences such as (4) correspond to atelic and semantically homogeneous events, our analysis can support inferences from sentences such as (26) to sentences such as (27)

(26) United Flight 342 has moved toward Logan Airport for the last fifteen minutes.
(27) United Flight 342 moved toward Logan Airport ten minutes ago.

Inference patterns between sentences such as (26) and (27) are, in fact, highly relevant for data base interface systems that process spatial information. Imagine that sentence (26) is presented to a database that monitors plane movements. If the system does not have the capability to infer that the event described in (26) is true at any subinterval of the fifteen minutes mentioned in (26), the United flight in question would erroneously not be counted when the answer to a subsequent query such as (28) is computed.

(28) How many planes moved toward Logan Airport ten minutes ago?

If we compare the semantics of *toward*, as we have defined it in (18) above with the semantics of *to* as defined in (11), it turns out that *to* is heterogeneous in its reference in the same way as accomplishments. Recall that the semantics of *to* is defined in terms of a complete path between a point of origin and a point of destination. Since for any given path there do not exist any sublocations within that path that themselves would qualify as a path between the same two locations, the heterogeneous reference property of *to* follows automatically.

5.5 Conclusion

In order to make an even stronger case in favor of my analysis of directional modifiers, I would have to demonstrate how it can be generalized to locative prepositions other than *to* and *toward*. Even though we cannot discuss this issue in detail in the present paper, I should like to point out in conclusion that the notion of a PATH plays an important role in the treatment of other directional prepositions such as *between*, *along* and *across*. In the case of *across* the path seems to be bounded by two locations on the peripherie of the referent of the NP *across* is combined with, i.e. *across the meadow* specifies some path extending from one end of the meadow to the

other. Notice, however, that the two locations that mark the two endpoints of such a path cannot be chosen arbitrarily but in some sense have to be "opposite each other". Undoubtedly, various pragmatic considerations enter the picture if one wants to make this requirement of *oppositeness* formally more precise. Thus, it appears that the notion of a path has to be complemented by additional constraints, if one wants to account for semantically more complex prepositions such as *across*. Even though we will have to leave the formulation of such additional constraints to future research, it should be obvious from these brief remarks that the notion of a path is a central notion for the semantics of directional modifiers in general.

REFERENCES

Carlson, Gregory N (1977). Reference to Kinds in English. University of Massachusetts dissertation

Celce-Murcia, M (1972) Paradigms for Sentence Recognition. Technical Report HRT-15092 790' System Development Corporation Santa Monica, CA.

Cresswell, Maxwell (1978) 'Prepositions and Points of View' Linguistics and Philosophy Vol 2 1, pp. 1-41

Davidson, Donald (1967). 'The Logical Form of Action Sentences'. In. Rescher, Nicholas ed. The Logic of Decision and Action. Pittsburgh. University of Pittsburgh Press, pp. 81-95.

Gunji, Takao and Norm Sondheimer (1980). 'The Mutual Relevance of Model-theoretic Semantics and Artificial Intelligence'. SMIL. The Journal of the Linguistic Calculus Vol 3, pp. 5-42.

Hendrix, Gary, Craig Thompson, and Jonathan Slocum (1973). 'Language Processing via Canonical Verbs and Semantic Models'. Proceedings of IJCAI-73.

Hinrichs, Erhard (1985). A Compositional Semantics for Aktionsarten and NP Reference in English Ph.D. dissertation, Ohio State University.

Montague, Richard (1970). 'Universal Grammar'. Theoria 36, pp 373-398.

Montague, Richard (1973). 'The Proper Treatment of Quantification in Ordinary English'. In. Hintikka, J., J. Moravcsik, and P. Suppes eds. Approaches to Natural Language. Reidel. Dordrecht.

Reichenbach, Hans (1947). Elements of Symbolic Logic. Berkeley University of California Press

Schank, Roger (1975) Conceptual Information Processing. North-Holland New York

Vendler, Zeno (1967) Linguistics in Philosophy. Ithaca Cornell University Press

6. REFERENCE AND REFERENCE FAILURES

Bradley A. Goodman

Abstract

The goal of this work is the enrichment of human-machine interactions in a natural language environment. Because a speaker and listener cannot be assured to have the same beliefs, contexts, perceptions, backgrounds, or goals at each point in a conversation, difficulties and mistakes arise when a listener interprets a speaker's utterance. These mistakes can lead to various kinds of misunderstandings between speaker and listener, including reference failures or failure to understand the speaker's intention. We call these misunderstandings *miscommunication*. Such mistakes can slow, and possibly break down, communication. Our goal is to recognize and isolate such miscommunications and circumvent them. This paper highlights a particular class of miscommunication - reference problems - by describing a case study and techniques for avoiding failures of reference. We want to illustrate a framework less restrictive than earlier ones by allowing a speaker leeway in forming an utterance about a task and in determining the conversational vehicle to deliver it. This paper also promotes a new view for extensional reference.

6.1 Introduction

Reference in the real world differs greatly from the reference processes modelled in current natural language systems. A speaker in the real world is a rational agent who must make a decision about his description in a limited time, with limited resources, knowledge, and abilities. In particular, the speaker's perceptual and communicative skills are imperfect or his model of the listener is erroneous or incomplete. Additionally, a speaker can also be sloppy in his description. Since the speaker's goal in the reference process is to construct a description that "works" for the listener, the listener, from his viewpoint, must take these imperfections into account when trying to interpret the speaker's utterances. Yet, listeners, too, have imperfect perceptual or communicative skills and can be sloppy. Hence, they must be prepared to deal with their own imperfections when performing reference identification. In real reference, listener's often recover from initial misunderstandings with or without help from the speaker. Natural language understanding systems must do this, too. Therefore, in performing the reference process, a system should assume and expect problems.

The focus of my work in [3, 4, 5] was to study how one could build robust natural language processing systems that can detect and recover from miscommunication. I investigated how people communicate and how they recover from problems in communication. That investigation centered on reference problems, problems a listener has determining whom or what a speaker is talking about. A collection of protocols of a speaker explaining to a listener how to assemble a toy water pump were studied and the common errors in speakers' descriptions were categorized. The study led to the development of techniques for avoiding failures of reference that were employed in the reference identification component of a natural language understanding program.

The traditional approaches to reference identification in natural language systems were found to be less flexible than people's real behavior. In particular, listeners often find the correct referent even when the speaker's description does not describe any object in the world. To model a listener's behavior, a new component was added to the traditional reference identification mechanism to resolve difficulties in a speaker's description. This new component uses knowledge about linguistic and physical context in a negotiation process that determines the most likely places for error in the speaker's utterance. The actual repair of the speaker's description is achieved by using the knowledge sources to guide relaxation techniques that delete or replace portions of the description. The algorithm developed more closely approximates people's behavior than reference algorithms designed in the past. The next section describes in more detail my work on reference.

6.2 Reference

Communication involves a series of utterances from a speaker to a hearer. The hearer uses these utterances to access his own knowledge and the world around him. Some of these utterances are noun phrases that refer to objects, places, ideas and people that exist in the real world or in some imaginary world. They cannot be considered in isolation. For example, consider the utterance "Give me that thing." It can be uttered in many different situations and can result in different referents of "that thing." Understanding such referring expressions requires the hearer to take into account the speaker's intention, the speaker's overall goal, the beliefs of the speaker and hearer, the linguistic context, the physical context, and the syntax and

semantics of the current utterance. The hearer could misinterpret the speaker's information in any one of these parts of communication. Such misunderstandings constitute miscommunication. In my research I focused primarily on effects of the linguistic context and the physical context.

To explore such reference problems, the following method was devised and followed. First, protocols of subjects communicating about a task were analyzed. Knowledge that people used to recover from reference miscommunications – knowledge about the world and about language – was then isolated. Algorithms were designed to apply a person's knowledge about linguistic and physical context to determine the most likely places for error in the speaker's utterance. Then, computer programs were written: (1) to represent a spatially complex physical world, (2) to manipulate the structure of that representation to reflect the changes caused by the listener's interpretation of the speaker's utterances and by physical actions to the world, (3) to perform referent identification on noun phrases, and, when referent identification failed, (4) to search the physical world for reasonable candidates for the referent. These programs form one component of a natural language system.

One goal in this summary of my research is to illustrate how my views on reference identification departed from views held by other researchers in artificial intelligence. Another goal is to show where my research fits in the scheme of natural language understanding by computers. My last goal is to summarize the approach of my research.

6.3 A new reference paradigm from a computational viewpoint

Reference identification is a search process where a listener looks for something in the world that satisfies a speaker's uttered description. A computational scheme for performing such reference identifications has evolved from work by other artificial intelligence researchers (e.g., see [6]). That traditional approach succeeds if a referent is found, or fails if no referent is found (see Figure 6-1(a)). However, a reference identification component must be more versatile than those previously constructed. The excerpts provided in [3] show that the traditional approach is inadequate because people's real behavior is much more elaborate. In particular, listeners often find the correct referent even when the speaker's description does not describe any object in the world. For example, a speaker could describe a turquoise

block as the "blue block." Most listeners would go ahead and assume that the turquoise block was the one the speaker meant since turquoise and blue are similar colors.

A key feature to reference identification is "negotiation." Negotiation in reference identification comes in two forms. First, it can occur between the listener and the speaker. The listener can step back, expand greatly on the speaker's description of a plausible referent, and ask for confirmation that he has indeed found the correct referent. For example, a listener could initiate negotiation with "I'm confused. Are you talking about the thing that is kind of flared at the top? Couple inches long. It's kind of blue." Second, negotiation can be with oneself. This self-negotiation is the one that I was most concerned with in this research. The listener considers aspects of the speaker's description, the context of the communication, the listener's own abilities, and other relevant sources of knowledge. He then applies that deliberation to determine whether one referent candidate is better than another or, if no candidate is found, what are the most likely places for error or confusion. Such negotiation can result in the listener testing whether or not a particular referent works. For example, linguistic descriptions can influence a listener's perception of the world. The listener must ask himself whether he can perceive one of the objects in the world the way the speaker described it. In some cases, the listener's perception may <u>overrule</u> parts of the description because the listener can't perceive it the way the speaker described it.

To repair the traditional approach I developed an algorithm that captures for certain cases the listener's ability to negotiate with himself for a referent. It can search for a referent and, if it doesn't find one, it can try to find possible referent candidates that might work, and then loosen the speaker's description using knowledge about the speaker, the conversation, and the listener himself. Thus, the reference process becomes multi-step and resumable. This computational model, which I call "FWIM" for "Find What I Mean", is more faithful to the data than the traditional model (see Figure 6-1(b)).

One means of making sense of a failed description is to delete or replace the portions that cause it not to match objects in the hearer's world. In my program I am using "relaxation" techniques to capture this behavior. My reference identification module treats descriptions as approximate. It relaxes a description in order to find a referent when the literal content of the description fails to provide the needed

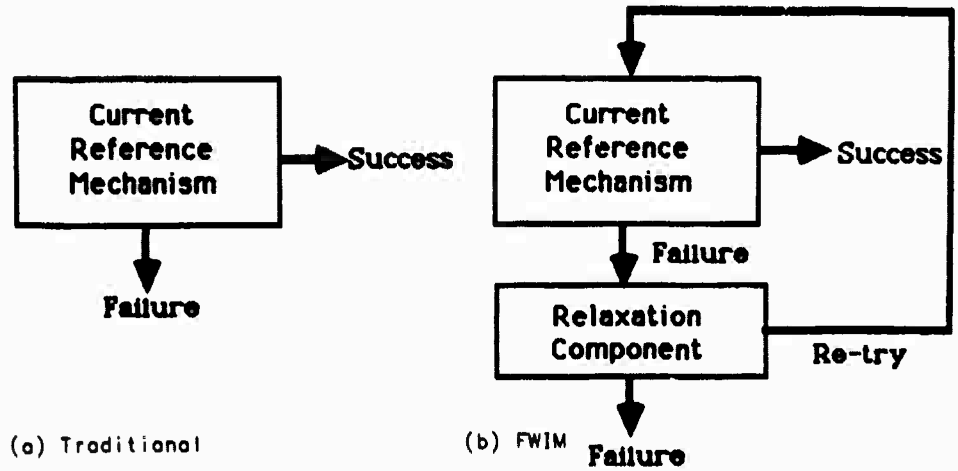


Figure 6-1: Approaches to reference identification

information. Relaxation, however, is not performed blindly on the description. I try to model a person's behavior by drawing on sources of knowledge used by people. I have developed a computational model that can relax aspects of a description using many of these sources of knowledge. Relaxation then becomes a form of communication repair (in the style of the work on repair theory found in [1]). A goal in my model is to use the knowledge sources to reduce the number of referent candidates that must be considered while making sure that a particular relaxation makes sense. A brief description of it follows.

The component works by first selecting with a partial matcher a set of reasonable referent candidates for the speaker's description (see also [7]). The candidates are selected by searching the knowledge base, scoring partial matches of each candidate to the speaker's description, and selecting those with higher scores. The component then generates, using information from the knowledge sources, a relaxation ordering graph that describes the order to relax features in the speaker's description. Finally, it combines the candidates with the ordering to yield the most likely referent. An ordered relaxation of parts of the speaker's description can be provided by consulting knowledge known about linguistics (the actual form of the speaker's utterance), perception (physical aspects of the world and the listener's ability to distinguish different feature values in that world), specificity (hierarchical knowledge to judge how vague or specific a particular feature value is), and others. In other words, the algorithm attempts to show how a listener might judge the importance of the features specified in a speaker's description using knowledge about linguistic and physical context. Figure 6-2 illustrates this process. The speaker's description is represented at the top of the figure. The set of specified features and their assigned feature value (e.g., the pair Color Maroon) are also shown there. A set of objects in the real world are selected by the partial matcher as potential

candidates for the referent. These candidates are shown near the top of the figure (C_1, C_2, ..., C_n). Inside each box is a set of features and feature values that describe that object. A set of partial orderings are generated that suggest which features in the speaker's description should be relaxed first – one ordering for each knowledge source (shown as "Linguistic," "Perceptual," and "Hierarchical" in the figure). For example, linguistic knowledge recommends relaxing Color or Shape before Function, and relaxing Function before Size. A control structure was designed that takes the speaker's description, puts all the (partial) orders together, and then attempts to satisfy them as best it can. This is illustrated at the bottom of the diagram by the reordered referent candidates.

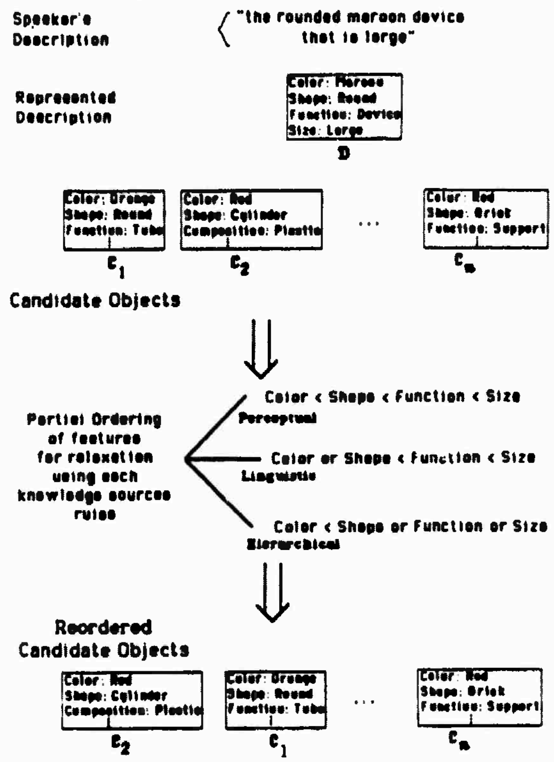


Figure 6–2: Reordering referent candidates

6.4 Summary

My goal in this work is to build robust natural language understanding systems, allowing them to detect and avoid miscommunication. The goal is not to make a perfect listener but a more tolerant one that could avoid many mistakes, though it may still be wrong on occasion. In this summary of my research, I indicated that problems can occur during communication. I showed that reference mistakes are one kind of obstacle to robust communication. To tackle reference errors, I described how

to extend the succeed/fail paradigm followed by previous natural language researchers.

I represented real world objects hierarchically in a knowledge base using a representation language, NIKL, that follows in the tradition of semantic networks and frames. In such a representation framework, the reference identification task looks for a referent by comparing the representation of the speaker's input to elements in the knowledge base by using a matching procedure. Failure to find a referent in previous reference identification systems resulted in the unsuccessful termination of the reference task. I claim that people behave better than this and explicitly illustrated such cases in an expert-apprentice domain about toy water pumps [3].

I developed a theory of relaxation for recovering from reference failures that provides a much better model for human performance. When people are asked to identify objects, they appear to behave in a particular way: find candidates, adjust as necessary, re-try, and, if necessary, give up and ask for help. I claim that relaxation is an integral part of this process and that the particular parameters of relaxation differ from task to task and person to person. My work models the relaxation process and provides a computational model for experimenting with the different parameters. The theory incorporates the same language and physical knowledge that people use in performing reference identification to guide the relaxation process. This knowledge is represented as a set of rules and as data in a hierarchical knowledge base. Rule-based relaxation provided a methodical way to use knowledge about language and the world to find a referent. The hierarchical representation made it possible to tackle issues of imprecision and over-specification in a speaker's description. It allows one to check the position of a description in the hierarchy and to use that position to judge imprecision and over-specification and to suggest possible repairs to the description.

Interestingly, one would expect that "closest" match would suffice to solve the problem of finding a referent. I showed, however, that it doesn't usually provide you with the correct referent. Closest match isn't sufficient because there are many features associated with an object and, thus, determining which of those features to keep and which to drop is a difficult problem due to the combinatorics and the effects of context. The relaxation method described circumvents the problem by using the knowledge that people have about language and the physical world to prune down the search space.

6.5 Future directions

The FWIM reference identification system I developed models the reference process by the classification operation of NIKL. I need a more complicated model for reference. That model might need a complete identification plan that requires making inferences beyond those provided by classification. The model could also require the execution of a physical action by the listener before determining the proper referent. Cohen gives two excellent examples of such reference plans (pg. 101, [2]). The first, "the magnetic screwdriver, please," requires the listener to place various screwdrivers against metal to determine which is magnetic. The second, "the three two-inch long salted green noodles" requires the listener to count, examine, measure and taste to discover the proper referent.

ACKNOWLEDGEMENTS

This research was supported in part by the Center for the Study of Reading under Contract No. 400-81-0030 of the National Institute of Education and by the Advanced Research Projects Agency of the Department of Defense under Contract No. N00014-85-C-0079.

I want to thank especially Candy Sidner for her insightful comments and suggestions during the course of this work. I'd also like to acknowledge the helpful comments of Marie Macaisa and Marc Vilain on this paper. Special thanks also to Phil Cohen, Scott Fertig and Kathy Starr for providing me with their water pump dialogues and for their invaluable observations on them.

REFERENCES

1. Brown, John Seely and VanLehn, Kurt. "Repair Theory. A Generative Theory of Bugs in Procedural Skills". *Cognitive Science 4*, 4 (1980), 379–426.

2. Cohen, Philip R. "The Pragmatics of Referring and the Modality of Communication". *Computational Linguistics 10*, 2 (April–June 1984), 97–146.

3. Goodman, Bradley A. *Communication and Miscommunication*. Ph.D. Th., University of Illinois, Urbana, Il., 1984. Also Report No.5681, BBN Laboratories Inc., Cambridge, Ma..

4. Goodman, Bradley A. Repairing Reference Identification Failures by Relaxation. Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics, Chicago, Illinois, July, 1985, pp. 204–217.

5. Goodman, Bradley A. "Reference Identification and Reference Identification Failures". *Computational Linguistics 12*, 4 (October–December 1986), 273–305.

6. Grosz, Barbara J. *The Representation and Use of Focus in Dialogue Understanding*. Ph.D. Th., University of California, Berkeley, Ca., 1977. Also, Technical Note 151, Stanford Research Institute, Menlo Park, Ca..

7. Joshi, Aravind K. A Note on Partial Match of Descriptions. Can One Simultaneously Question (Retrieve) and Inform (Update)? Theoretical Issues in Natural Language Processing–2, Urbana, Ill., July, 1978, pp. 184–186.

7. POINTING THE WAY: A UNIFIED TREATMENT OF REFERENTIAL GESTURE IN INTERACTIVE DISCOURSE

Erhard Hinrichs and Livia Polanyi

Abstract

In this paper, we argue that a complete model of interactively constructed natural discourse must provide a principled account of deictic gesture which establishes reference to non-linguistic objects, properties and relations. More specifically, we shall demonstrate that in order to account for the contextual relevance of linguistic units such as words, phrases and sentences, an adequate discourse model must include (1) a compositional syntax and semantics at the sentence level which is capable of dealing with fragmentary linguistic input and (2) a discourse component which accepts deictic gestures along with traditional linguistic units as input and assigns the correct context of interpretation to each structure parsed.

7.1 Introduction

The necessity of including an analysis of non-verbally encoded information became clear to us when we started analyzing a corpus of Spatial Planning Protocols collected for the purpose of analyzing how actual speakers interactively construct plans. We soon discovered that failure to take non-verbal information into account resulted in misunderstanding the nature of the communication being analyzed. Repeatedly, we found that we did not understand what was going on in our data from examining audio material alone. We needed the videotapes of the protocol collection sessions to provide us with vital information about what was, in fact, happening in the interaction between our research subjects.

The protocol collection sessions involved playing a game called "Travelling through Europe". Two subjects playing together against a researcher were given a set

of nine European cities and a game board which consists of a map of Europe marked with over one hundred city names joined together by lines representing legal routes. The task of the subjects was to plan the most efficient route -- one which would allow them to visit all nine cities on their itinerary in the smallest number of steps. "Playing the game" involved planning an itinerary and then taking turns throwing a die and moving a marker on the board the number of city steps corresponding to the number shown on the die. Updating and changing plans was allowed at any time.

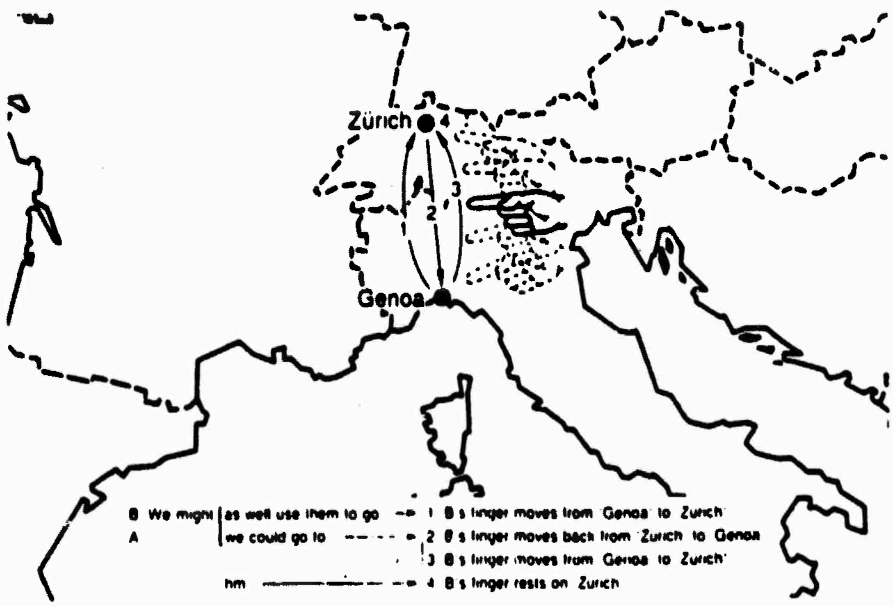
7.2 Semantic Interpretation of Gesturally Suppleted, Verbally Incomplete Verbally Incomplete Propositions

Consider the piece of discourse in (1), an example taken from the corpus of spatial planning protocols. Without the accompanying pointing gestures made by B, which in (1) are set off in bold-face and by curly brackets, we might well characterize B's functioning in this piece of discourse as inarticulate and indecisive.

(1) A. we have two points left.
B. OKAY.
So [we can go to
A. [We might as well use them
to go. }}**B's finger at Genoa**{{
}}**B's finger moves from piece at Genoa to Zurich.**{{
B. We could go to ...
h..m = }}**hand off Zurich**{{

A reasonably correct analysis of this data is only possible when the non-verbal information is taken into account. When B's gestures are considered part of the signifying mechanism he is employing, it becomes clear that B, far from producing "incomplete" proposition carrying units and adding little to the planning process, is actively suggesting a very definite course of action. He is proposing that the players should choose a route which takes them from "Genoa" to "Zurich".

B's Gesture



On the observational level the example demonstrates the importance of integrating linguistic and non-linguistic information for the process of (computational) discourse understanding since without the gestural information the discourse is semantically incomplete. For a theory of discourse understanding the example raises at least the following issues.

1. An adequate discourse parser has to be able to accept as input elliptical sentences such as *We could go to* in our example and augment the semantic interpretation of such sentence fragments with the semantic interpretation of referential gestures which supplete the verbal part of such utterances.

2. Once an elliptical sentence fragment in combination with its gestural suppletive has been correctly parsed, interpretive structures are needed to determine what function such a unit of discourse plays for the discourse as a whole.

We will argue in this paper that a discourse model which is based on the *Lingustic Discourse Model* developed in Polanyi and Scha (1984) and Polanyi (1985) can provide a principled account of how to provide a compositional syntax and semantics

for individual sentences of a discourse (including elliptical sentences) and how to assign meaning to individual sentences in the context of a discourse as a whole.

7.3 A Compositional Syntax and Semantics for Sentences in Discourse

Let us first consider that component of our discourse model that deals with the interpretation of individual sentences, in particular sentence fragments. We agree with a growing number of researchers both in the field of theoretical linguistics and in the field of artificial intelligence that the syntax/semantics interface of a grammar should account for the data in terms of a grammar formalism that is as restrictive as possible in its generative power and hence in its computational complexity. At the same time we take it to be the fundamental task of a semantic theory of natural language to relate in a systematic fashion linguistic expressions to real-world objects, relations and states-of-affairs. In particular, we follow Richard Montague (cf. Montague 1970, 1973) and others in assuming that the meaning of sentences should be given at least partly in terms of the conditions that would make them count as true in a given world or in some state of affairs. Moreover, we assume with Montague and others that the semantic composition of any linguistic expression is to be conceived of as a homomorphic image of its syntactic composition. That is, we assume Montague's interpetation of Frege's Principle which says that the meaning of a syntactically complex expression has to be derived in terms of the meanings of its syntactic parts.

For the syntactic analysis of individual sentences we adopt for a our discourse model the version of categorial grammar developed by Steedman (1985) and by Steedman and Ades (1983). Our decision to adopt their syntactic framework is motivated by the following considerations. Categorial grammars are restrictive in their generative capacity. As shown by Friedman et al (1985), depending on the type of syntactic functors permitted, categorial grammars of the type Steedman (1985) discusses generate at best only context-free languages and at worst only the set of context-sensitive languages. Moreover, as Steedman himself points out, his categorial grammars provide a natural mechanism for left-to-right parsing of an input stream,

which is particularly attractive for computational purposes.[1] Finally, and most relevant for the purposes of this paper, Steedman's version of categorial grammar offers a principled treatment of syntactically incomplete utterances such the elliptical *We could go to* in the discourse example that we are analyzing in this paper

Relying crucially on Steedman's notion of functional composition, we assign the parse in (2) to the elliptical utterance *We could go to* which in our example is suppleted by a referential gesture.

```
(2)  We        could      go    to
     _____     _______    __    ________
     S/FVP     FVP/VP     VP    VP/VP/VP

     ______________ (FPC)  ______________ (FPC)
     S/VP                  VP/NP
     ___________________________________ (FPC)
                  S/NP
```

Apart from the usual operation of functional application or Forward Combination (FC) in a categorial grammar, which can be stated as in (3), Steedman and Ades allow for operations of partial combination or functional composition. Among such partial combination operators, they define Forward Partial Combination (FPC) as in (4).

(3) X/Y Y --> X (FFA)

(4) X/Y Y/Z --> X/Z (FPC)

It is the operation of Forward Partial Combination that allows us first to combine the verb *go* in (2) with its prepositional modifier *to*, even though the preposition is lacking its NP argument and then to combine the resulting phrase *go to* with the remaining words in the sentence.

Let us consider next how elliptical sentence as in (2) can be interpreted together with their accompanying deictic gestures. For the semantic interpretation of

[1]For the use of extended categoriol grammar for parsing natural language see also Wittenburg (1986).

space and time, which plays a central role in a task domain such as spatial planning, we adopt the approach to spatial and temporal information which I have developed in my dissertation, Hinrichs (1985). In particular, we adopt from this earlier work the semantics of motion verbs and directional modifiers which is needed for the interpretation of (2). We treat motion verbs as *stage level predicates* in the sense of Carlson (1977), namely as predicates whose arguments refer to stages of individuals, which are interpreted as the space-time locations occupied by the individuals. Following Carlson, individuals are connected to their stages in terms of a realization relation R, which associates a given individual with all of the (spatio-temporal) locations at which that individual is present. Motion verbs such as *go* are interpeted in terms of two-place predicates whose first argument position ranges over individual stages which realize the referent of the subject NP combining with *go*. The rightmost argument position ranges over event stages realizing the event that the referent of the subject NP is engaged in. We thus adopt the strategy of Davidson (1967) to reserve in the argument structure of action verbs an extra argument position which refers to events, or more specifically event stages. Directional prepositions such as *to* function semantically as modifiers which specify further the location at which the event associated with an action verb takes place. In particular, the translation of *to* into the interpretation language introduces the notion of a spatio-temporal path between some implicit point of origin, denoted by the free variable l_r occuring in the formula in (5) and some point of destination, which is represented by the referent of the NP that *to* combines with. Given this analysis of *go* and its directional modifier *to*, the elliptical sentence in (2) is translated into the formula in (5).[2]

$$(5)\ \lambda P\, P\, [\, \lambda y^o \diamond \exists e^i, l_1, l_2, l_3\, [\, R(l_1, y^o)\ \&\ R(l_2, we')\ \&\ R(l_3, e^i)\ \&\ go^+(l_2, l_3)\ \&\ PATH\,(l_3, l_r, l_1)\,]\,]$$

The variable P in (5) is meant to range over properties of properties of individuals. Thus, the formula denotes the set of such complex properties of some object denoted by y^o whose space-time location l_1 forms the potential destination of some path l_3, which originates at some reference point l_r and which is traversed during an event of some individuals denoted by *we* going from l_r to the location of y^o.

[2]The $\diamond$ symbol in (5) is meant to represent the propostional operator of epistemic possibility, as standardly used in modal logic. The meaning of this operator is most closely represented in English by the modal *can*, rather than *could*. Hence, the translation in (5) somewhat oversimplifies the counterfactual modal force of (2).

If (5) represents the translation of (2), the deictic gestures accompanying the elliptic utterance can be interpreted in the following way. the formula in (5) is deficient in two respects. the reference point r has to be specified and the object y has to be identified which has among its properties that it represents the end-point of some path originating at l_r. The deictic gestures identify Genoa and Zurich as the two objects representing the reference point and the point of destination, respectively. Since the property that "ties these two objects together" in (5) is the notion of a path and since both objects represent space-time locations, the most salient interpretation for the gesture is that it indicates a path between the two cities. Combining the referents of the deictic gestures with the formula in (5), we arrive at the translation in (6)

(6) $\Leftrightarrow \exists\, e^i, l_1, l_2, l_3$ [R(l_1,Zurich') & R(l_2,we') & R(l_3,e^i) & go$^+$(l_2,l_3) & PATH (l_3,$l_{Genoa'}$,l_1)]

In the context of the game, the potential path identified by the two players does, of course, not stand for a real-world journey between two cities in Switzerland, but rather for a particular move in a game of route optimization. The conversational function of the elliptical utterance can be explicated in terms of the Linguistic Discourse Model (LDM) model by providing the appropriate elements of discourse grammar relative to which the utterance has to be interpreted.

7.4 The Conversational Function of the Elliptical Utterance

To elucidate how interpretive contexts may be arrived at, we shall make use of the Linguistic Discourse Model - a theory of discourse structure which provides theoretical notions and formal machinery necessary to assign B's gesturally suppleted utterance to the relevant contexts. The LDM, a theory of the structural and semantic relations obtaining among clauses in discourse, is formulated as a Discourse Parser. The unit of input to the parser is normally a linguistically realized clause encoding a single proposition, but may also be a non-verbally suppleted utterance such as (1) or a non-verbal action sequence such as the pointing gesture in (14) below.

Before beginning the LDM analysis, let us first consider briefly some of the factors which we intuitively take into account in interpreting B's proposition as a

proposal that a route from Genoa to Zurich be taken in the game which they are playing.

- o A and B are engaged in an interaction with each other.
- o They constitute a team playing the "Game Travelling through Europe" as part of an experiment
- o A and B play this game cooperatively. They agree together to moves which are acceptable to both and which they believe to be permitted by the rules of the game.
- o It is A and B's turn in the game.
- o After the die is thrown and it is clear how many points are available to them. A and B have to agree upon a course of action.
- o Agreeing upon a course of action involves a negotiation in which proposals and counterproposals are made.
- o Putting some course of action on the table, is a possible first step in a negotiation sequence.
- o The presence of the modal *might* in the verbally encoded *We might as well use them to go to* signals proposal to perform the action specified in the embedded phrase.
- o The verbally encoded phrase is suppleted by B's use of his finger to connect two dots on the gameboard construed in the game as representing" cities".
- o The beginning point of B's tracing motion is at the dot marked "Genoa" and the trace ends at the dot marked "Zurich". A&B's playing token is located at "Genoa" as the turn begins. The number of steps to "Zurich" is two, which is the number thrown on the die a moment earlier.

The LDM provides a formal mechanism for capturing these intuitive conceptions of what is happening at the time of B's gesturally suppleted utterance

7.5 The LDM

The LDM parser describes a discourse as built up by means of a sequencing and recursive embedding of discourse constituents The Parser proceeds on a clause by clause left to right basis through the discourse assigning each constituent clause to appropriate interpretive contexts and assigning to the discourse as a whole an incrementally constructed structural description.

In real language use, utterances occur in real or modelled Interactions and are used to carry out various interactive business, so called Speech Events (Hymes 1972). In the LDM which models language structure at the discourse level, there are similarly no uncontextualized clauses. Clauses in discourse are linked together in discourse constituent units (dcu's) of one or more clauses and used to build up the genre units - here called "Discourse Units' - which speakers use to develop conventionally structured semantically coherent accounts of the states of affairs in some Modelled Discourse World

Under an LDM analysis, the propositional information encoded in a given "clause" receives its interpretation relative to a hierarchy of contexts deriving from units of the several types (dcu, DU, Speech Event and Interaction).Speech Events are further divided into "Moves". Each unit type is further subdivided into sub-types of various sorts.[3] Each unit type has an associated grammar which specifies its legal constituents and their permissable orders.[4] Grammars for the Turn and Negotiation Sequence are discussed below. (See figures 7 and 8 below.)

In assigning a structural description to a developing discourse on a clause by clause left to right basis, the LDM makes use of these grammars together with semantic and structural information about the propositional content and contexts of occurrence of the given clause. The required information about the clause's content derives ultimately from the proposition it encodes, while contextual information reflects specific higher level contextualizing units in which the incoming clause participates.

---

[3]The "clause" is a one element dcu. "Sequential", "and expansion" dcu's are the two most important classes of complex dcu. The "story " "argument" and "proposal" are relatively common DU's, while there are Speech Event units as various as conversations, consultations with the village elder, bridge parties, and games of bridge.

[4]Often these grammars are simple phrase structure grammars, although more complex grammars are also needed to capture the structure of some types of units.

7.6 Structural Relations in Discourse

Discourse constituents are related structurally to one another by coordination and subordination The LDM is fully recursive and constituents of all types are legally embeddable within one another. Interruptions, elaborations, asides and parentheticals are uniformly treated as discourse subordinations because they disturb the orderly development of some ongoing discourse activity. Discourse Coordination is more constrained and is effectively limited to Topic chains, narratives, and other list dcu structures as well as sequences of moves in Discourse Units and Speech Events and sequences of independent Interactions .

7.7 Discourse Parsing with the LDM

In order to process a discourse, the discourse parser makes use of an inventory of individual Type and sub-Type Grammars, calling upon parsers associated with them as needed to process constituents of different sorts. There is no limit to the number of times an individual parser might be made use of to process a given discourse, nor is there any constraint placed on the order in which those parsers must be called. The frequency of calls to any parser and the order of calls depends entirely on the nature of the individual discourse.

The LDM Parser builds the Discourse History Parse Tree for a given discourse by coordinating and subordinating incoming clauses into discourse constituent units at existing or newly created rightmost nodes in the tree Only rightmost nodes are structurally accessible.

By comparing the content and discourse unit contexts of the incoming unit with the information available at open nodes, the Parser determines whether (1) to coordinate the target utterance to the immediate preceding constituent as a sister node, (2) to add it to the tree as a rightmost sister to some higher level constituent or,(3) to subordinate it to an existing accessible constituent available somewhere in

the tree.[5]

The LDM framework thereby resolves an apparently insoluble problem in discourse analysis. Anything can happen in any discourse and therefore a theory of discourse structure must account for the highly individual and possibly unexpected structure of any given discourse while, simultaneously, accounting for the fact that, at any given moment, speakers are normally quite clear about the kind of discourse activity underway and have very definite expectations about what is likely to happen next.

These expectations about how the discourse will proceed captured in the grammars of the various unit types are exploited in the LDM's assigning to B's utterance the status of a PROPOSAL.

7.8 Analyzing the Discourse Context of B's Utterance

When B's utterance is encountered by the parser, it has just finished dealing with the previous utterance and has assigned to *We have two points left* a set of interpretive contexts reflecting its current state. These contexts, shown in (7), are occasioned by the throw of the die during one of A&B's turns at Play in the Speech Event$_{\text{Playing the game "Travelling Through Europe"}}$ which is itself part of a Speech Event$_{\text{Experiment}}$ taking place during a unique spatio/temporal/social

[5]Coordination to a previous unit located at a still accessible node in the Discourse Tree is permitted only if the two units are constituents of the same set of higher level units and then only if the Grammar of the lowest level common unit specifies that the incoming unit can legally function as a next constituent. Subordination is the default structural relation and obtains (1) between an incoming clause and a structurally accessible dcu if the values associated with the propositional content of the clause bear an instantiation relation to (at least a subset of) the semantic context values of the candidate mother node or (2) between a clause and the last unit parsed if the semantic values associated with the propositional value of the clause have no semantic relation with the semantic values at any available node. (See section 10. below) For the purposes of the present analysis, we will be concerned only with discourse coordination.

Interaction$_{\text{Kaplan Contexts}_a}$[6]

(7) <Interaction$_{\text{Kaplan Contexts}_a}$
<Speech Event$_{\text{Experiment}}$
<Speech Event$_{\text{Playing Travelling through Europe}}$
<Move$_{\text{Take turns}}$
<Sub-Move$_{\text{Turn A\&B}}$
<Sub-Move$_{\text{Throw die}}$>>>>>>

According to the Grammar of A&B's Turn in (8),

(8) TURN GRAMMAR

Move$_{\text{Turn Team A\&B}}$----> Throw Die + <u>Negotiate Action</u> + Move Counter

the parser now expects A and B to <u>Negotiate</u> a course of action to take in deciding what "route" to use in accomplishing the part of their Game World Journey which would advance them towards their goal. According to the grammar of negotiation shown in (9), the first part of any <u>Negotiation Sequence</u> in this game is a <u>Proposal</u>, for what to do relative to the position of the players' piece on the map game board:

(9) NEGOTIATION GRAMMAR

Move$_{\text{Negotiation}}$----> <u>Make Proposal</u> + (Discussion of Proposal)
+ [Counter Proposal
+ (Discussion of Counter Proposal)]*
+ One Proposal Accepted

Expecting a Route Proposal, the parser easily processes B's gesturally suppleted utterance as such a Proposal since it conveys appropriate propositional information,

---

[6] Kaplan Contexts (see Kaplan ms.) specify real world temporal, spatial, and participant indices. "Kaplan context$_a$" is being used to identify the unique utterance situation which took place at a unique real world spatial Location$_a$ (Conference Room, BBN Labs, Cambridge, MA), at Temporal Index$_a$ (November 4, 1986, 10:00 - 10:15 AM), involving Participants$_{\text{A,B, E.H., L.P.}}$

and is encoded according to the syntactic conventions appropriate for signalling possibility. The parser assigns the formula in (6) the interpretive contexts shown in (10).

(10) <Interaction$_{\text{Kaplan Contexts}_0}$
<Speech Event$_{\text{Experiment}}$
<Speech Event$_{\text{Playing Travelling through Europe}}$
<Move$_{\text{Take Turns}}$
<Sub-Move Turn$_{\text{A\&B}}$
<Discourse Unit$_{\text{Negotiation of route to take}}$
<dcu$_{\text{Proposal}}$>>>>>>>

These contexts localize B's gesturally suppleted clause as a unique utterance relative to unique circumstances of utterance and are used by the LDM to compute how the encoding clause participates in the tree structure of the emerging discourse.

In order to assign, *We could go to*, or any other incoming clause a position in the Discourse History Parse Tree, the LDM parser attempts to match the contexts of the present utterance with those of the immediately preceding utterance *We have two points left* (shown above in 7).

These contexts are available in the tree of the developing discourse as the label at the node immediately dominating the terminal clause node as shown in Figure 11.

(11)

<1<2<3<4<TURN>>>>

<1<2<3<4<5<THROW DICE>>>>>>

"We have two points left"

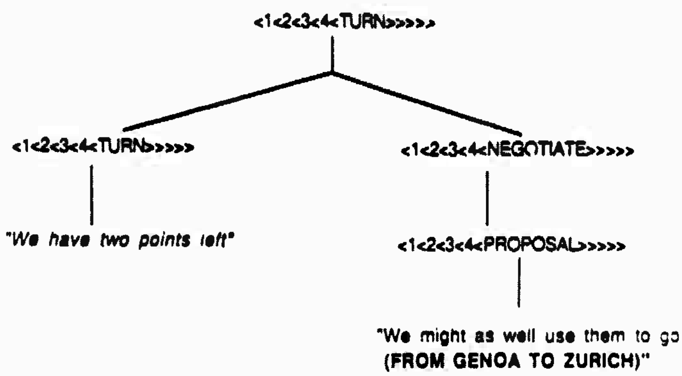
In the present case, therefore, the first five contexts match

- Interaction$_{\text{Kaplan Context}_0}$
- Speech Event$_{\text{Experiment}}$
- Speech Event$_{\text{Playing Travelling Through Europe}}$
- Move$_{\text{Complete Turns}}$
- Sub-Move$_{\text{Turn A\&B}}$

However, when processing Context 6 of *We could go to*, the parser is unable to effect a match. Context 6 of the preceding unit -- <Throw die> -- does not match Context 6 of the present clause which is <Negotiation of Route>. At this point, with reference to the state of the discourse as reflected in the parse tree, the parser must make use of the grammars of the discourse units currently under construction and the context information encoded at open nodes to decide whether to subordinate, coordinate, or superordinate the incoming unit at the node corresponding to Context 5 in the tree.

The decision process, in this case, is not complicated. Because the higher level interpretation contexts match and because "Negotiating a route to take" is an appropriate next constituent to follow "Throw Die" according to the Grammar of A & B's Turn, (see 7 above) *We could go to* is coordinated with *We have two points left* under a coordination node carrying the values of the five matched contexts as illustrated in Figure 12.

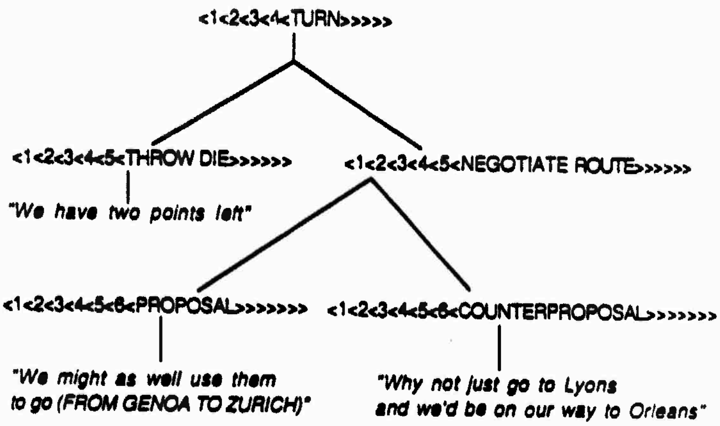
(12)



Carrying the analysis one step further, we can now account for the relevance of A's next remark, *why not just go to Lyons and we'd be on our way to Orleans?* In the context where it occurs, A's comment is commonplace and fully coherent. "*Lyons* is seen as a counterproposal to B's gesturally communicated proposal to follow a route from *Genoa* to *Zurich*.

Viewed in terms of the LDM framework, A's behavior is predictable from the Grammar of Negotiation. Following B's Proposal, A makes the next Move allowed by that Grammar and utters a complex clausal construction which functions in the ongoing context as a counter-proposal to B. Since contexts 1-6 of the two utterances are the same, as is shown in Figure 13, the LDM when processing *Why no just go to Lyons and we'd be on our way to Orleans.*, will eventually coordinate it to *We could go to* [*Zurich from Genoa*] under a node with values <1-6> on the Discourse History Parse Tree.

(13)



10. ESTABISHING THE CONTEXTUALLY CONSTRAINED INTERPRETATION OF B'S UTTERANCE

In order to establish the contextually constrained interpretation to be accorded a given unit in a discourse, each unit under an LDM analysis is associated with a semantic frame which gives relevant semantic information about the unit in terms of a number of semantic indices which specify parameters such as temporal, spatial, goal and participant information abstracted from the unit's content.

Formally defined operations on the semantic parameters of low level constituents yield the parameters for higher level constituents which contextualize them. Lower level constituents may only participate in high level constituents if the semantic values of at least a subset of the lower level unit are related systematically to the values of a higher level unit located at an open node on the tree. (Polanyi 1985).

The semantic parameters of purely linguistic units (clauses, dcu's, and Discourse Units) are set relative to the interpretation accorded a unit's propositional content.

For Interactions the indices in the semantic frame are real world Kaplan Contexts as indicated below, while for Speech Events, the relevant semantic dimensions concern Speech Event roles, activities, and concerns. Temporal and spatial aspects of Speech Events are set relative to the activity at hand. Thus, at the Speech Event level, the same physical location used for an "Experiment" is defined differently from that same physical location used to hold an "Auction".

Therefore, the participant set (Player 1, Player 2, Player 3) of the

Speech Event$_{\text{Playing Travelling Through Europe}}$ is related systematically to a subset of the participants playing roles in the higher level Speech Event$_{\text{Experiment}}$ (Experimenter 1, Experimenter 2, Research Subject 1, Research Subject 2). The role playing participants of the Experiment Speech Event are similarly related to a proper subset of the participants of the Interaction $_{\text{Kaplan Context}_a}$ (A, B, E.H., L.P.) The indivdual A in the real world of the Interaction is defined relative to his Speech Event$_{\text{Experiment}}$ role as Research Subject$_1$ and to his role in Speech Event$_{\text{Playing Travelling Through Europe}}$ as Player 1.

Space and Time in the lower level units are likewise established with reference to the spatial and temporal parameters of higher level contexts. The Spatial parameter associated with the A&B's Turn, for example, is set relative to the Spatial parameters of the contextualizing higher level unit -- the Complete Turn. The Spatial parameters of the Complete Turn include all possible routes for both teams while the spatial parameters for Turn$_{A\&B}$ include only possible routes for A&B's Gameworld surrogate.

For the example in question, therefore, the possible interpretation of the spatial locations referred to in *We could go from "Zurich" from "Genoa"* is restricted by context computations to the *Genoa* and *Zurich* on the game board and cannot refer to the *"Genoa"* and *"Zurich"* in the real world, on any other map or relevant to any other world of discourse. *We* is similarly interpreted as *We the surrogates associated with We the Players associated with We A&B in the Real World in which the Interaction took place.*

7.9 A Gestural Proposal

Consider next the example in (14) in which B's gestures do not supplete a partially verbalized proposition but function independently to convey meaning.

(14) A. . . . a..nd then we've covered most of our ground
right there

{{B points at Berlin}}

We've still got Berlin **{{ B moves to Prague}}**

to hit. {{B's finger at Prague}}

B . Prague.

A . And some place called Lodz

If one has access only to the record of the spoken interaction, the talk seems somewhat incoherent at that point.[7] As was the case with example (1), in order to ascertain the coherence and relevance of the surrounding linguistic material correctly, sentential syntactic and semantic analyses must again be augmented by an understanding of the discourse context. When B's gestures are considered part of the signifying mechanism he is employing, it becomes clear that far from adding little to the planning, B is actively suggesting a very definite course of action. By pointing to Berlin and then to Prague, B makes nonverbal proposals -- suggesting that the players should include Berlin and then Prague on their itinerary. A's uttering "Berlin" and the "Prague" can now be understood as accepting B's proposal to include those cities in that order in their itinerary.

The LDM analysis of (14), B's pointing to Berlin on the game map is similar to the analysis of example 1. The point is interpretated as a proposal to visit Berlin after the cities A has specified. B's pointing gesture receives this interpretation because the pointing takes place as the discourse parser is processing a discourse unit of type PLAN (Linde and Goguen, 1978) and expects either (1) a reaction to A's proposal -- either accepting, rejecting or adding to it or (2) an initiation or resumption of an unrelated unit. In this structural position, deictic reference to a city known to be on the itinerary is interpreted as adding to the proposal a suggestion to visit that city next. B's point to Berlin thus functions as a proposal that Berlin be visited next.

---

[7]Lacking knowledge of B's non-verbal proposal, the researcher can no help but wonder where and how "Berlin" is encoded into A's cognitive map of Europe. A had been dealing systematically with geographical areas quite far removed from the center of Europe and "suddenly", with an intonational change, switched his attention to "Berlin". Without understanding that A's comment is a response to B's action it is not at all clear why A switches gears so suddenly or what principle of geographical organization he was using to structure his itinerary plan.

7.10 Conclusion

We have argued in this paper, then, that to be able to account for the full meaning and relevance of utterances in discourse a theory of discourse structure must be able to provide an account of both verball
y and non-verbally encoded information. We have shown how a formal treatment of sentential syntactic and semantic phenomena when augmented with a discourse model capable of assigning interpretive contexts to naturally occuring talk can be used to help us understand the full force of even a fragmentary utterance in a world of real world language use.

REFERENCES

Ades, A. E. and M. J. Steedman (1982). 'On the Order of Words'. *Linguistics and Philosophy* Vol. 4.4. pp. 517-558

Davidson, Donald (1967). 'The Logical Form of Action Sentences'. In. Rescher, Nicholas ed. The Logic of Decision and Action Pittsburgh. University of Pittsburgh Press, pp. 81-95.

Friedman, Joyce, Dawai Dai and Weiguo Wang (1986). *The Weak Generative Capacity of Parenthesis-Free Categorial Grammars* Boston University. Computer Science Tech Report # 86-001.

Hinrichs E (1985) *A Compositional Semantics of NP Reference and Aktionsarten in English*. Unpublished Ph.D. Dissertation. Ohio State University.

Hinrichs, E. and L. Polanyi (to appear). *Pointing the Way: Modelling Non-Verbal Referential Gesture in Natural Language Discourse*. BBN Technical Report. Bolt Beranek and Newman Labs: Cambridge.

Hymes, D. (1972). 'Models of the Interaction of Language and Social Life.' In. J. Gumperz and D. Hymes eds. Directions in Sociolinguistics. 35-71. New York. Holt, Renehart and Winston.

Kaplan, D. (ms.) 'Demonstratives. An Essay on the Semantics, Other Indexicals' (unpublished, University of California. Los Angeles).

Linde, C. and J. Goguen (1978). 'The Structure of Planning Discourse'. *Journal of Social and Biological Structure* Vol. 1, pp. 219-251.

Marslen-Wilson, W., E. Levy, and L.K. Tyler. 'Producing interpretable Discourse'. In Jarrella, R. and W. Klein eds. *Speech, Place, and Action*. Wiley and Sons. New York.

Montague, Richard (1970). 'Universal Grammar'. *Theoria*, Vol. 36, pp. 373-398.

Montague, Richard (1973). 'The Proper Treatment of Quantification in Ordinary English'. In: J. Hintikka, J. Moravcsik, and P. Suppes eds. *Approaches to Natural Language*. Reidel. Dordrecht.

Polanyi, L. and R. Scha (1984). 'A Syntactic Approach to Discourse Semantics'. *Proceedings of Coling84*. Stanford University. Stanford. pp. 413-419.

Polanyi, L. (1985). 'A Theory of Discourse Structure and Discourse Coherence'. *Proceedings of the 21st Meeting of the Chicago Linguistics Society*. Department of Linguistics, University of Chicago. Chicago, Illinois.

Steedman, M. J. (1985). 'Dependency and Coordination in the Grammar of Dutch and English' *Language*. Vol 61.3, pp. 523-568

Wittenburg, Kent (1986). *Some Properties of Combinatory Categorial Grammars of Relevance to Parsing*. MCC Technical Report Number. HI-012-86.

8. PUBLICATIONS

B. A. Goodman. "Reference Identification and Reference Identification Failures," *Computational Linguistics*, Vol 12, No 4, 1986 (a revised version also appears as *Rule-Based Relaxation of Reference Identification Failures*, Technical Report No. 396, Center for the Study of Reading, University of Illinois at Urbana-Champaign, 1986).

B. A. Goodman. "Repairing Reference Identification Failures by Relaxation," in Communication Failure in Dialogue and Discourse, Ronan Reilly (ed.), North-Holland, 1987 (also in *Proceedings of the 23rd Annual Meeting of the ACL*, July 1985).

B. A. Goodman, "Reference and Reference Failures," in Proceedings of Theoretical Issues in Natural Language Processing -3, TINLAP-3, New Mexico State University, Las Cruces, New Mexico, 1987 (also appears as Technical Report No. 398, Center for the Study of Reading, University of Illinois at Urbana-Champaign, 1986).

Abstract

The goal of this work is the enrichment of human-machine interactions in a natural language environment. Because a speaker and listener cannot be assured to have the same beliefs, contexts, perceptions, backgrounds, or goals at each point in a conversation, difficulties and mistakes arise when a listener interprets a speaker's utterance. These mistakes can lead to various kinds of misunderstandings between speaker and listener, including reference failures or failure to understand the speaker's intention. We call these misunderstandings *miscommunication*. Such mistakes can slow, and possibly break down, communication. Our goal is to recognize and isolate such miscommunications and circumvent them. These papers highlight a particular class of miscommunication - reference problems - by describing a case study and techniques for avoiding failures of reference. We want to illustrate a framework less restrictive than earlier ones by allowing a speaker leeway in forming an utterance about a task and in determining the conversational vehicle to deliver it. These papers also promotes a new view for extensional reference.

A. R. Haas, "A Syntactic Theory of Belief and Action," *Artificial Intelligence*, Vol. 28, No. 3, May 1986.

Abstract

If we assume that beliefs are sentences of first-order logic stored in an agent's head, we can build a simple and intuitively clear formalism for reasoning about beliefs. I apply this formalism to the standard logical problems about belief, and use it to describe the connection between belief and planning.

E. Hinrichs, "A Compositional Semantics for Directional Modifiers in English – Locative Case Reopened," *Proceedings of the 11th International Conference on Computational Linguistics*, pp. 347–349, August 1986.

Abstract

This paper presents a model-theoretic semantics for directional modifiers in English. The semantic theory presupposed for the analysis is that of Montague Grammar (cf. Montague 1970, 1973) which makes it possible to develop a strongly compositional treatment of directional modifiers. Such a treatment has significant computational advantages over case-based treatments of directional modifers that are advocated in the AI literature.

E. Hinrichs and L. Polanyi," Pointing the Way. A Unified Treatment of Referential Gesture in Interactive Discourse," *Proceedings of the 22nd Annual Meeting of the Chicago Linguistics Society*, 1986.

Abstract

In this paper, we argue that a complete model of interactively constructed natural discourse must provide a principled account of deictic gesture which establishes reference to non-linguistic objects, properties and relations. More specifically, we shall demonstrate that in order to account for the contextual relevance of linguistic units such as words, phrases and sentences, an adequate discourse model must include (1) a compositional syntax and semantics at the sentence level which is capable of dealing with fragmentary linguistic input and (2) a discourse component which accepts deictic gestures along with traditional linguistic units as input and assigns the correct context of interpretation to each structure parsed.

L. Polanyi, "The Linguistic Discourse Model. Towards a Formal Theory of Discourse Structure," BBN Technical Report No 6409, November 1986.

Abstract

Despite the apparent disfluency and disorganization of everyday talk, speakers all but flawlessly recover anaphoric references, interpret temporal and spatial deictic expressions and use language to shape utterances which demonstrate a clear and recoverable relationship to "the business at hand" in the talk and the contextualizing social setting. In the paper, we shall present a comprehensive formal model of discourse structure, the *Linguistic Discourse Model*, the *LDM*, which provides a uniform account of how speakers accomplish these tasks in constructing and understanding both maximally coherent and highly attenuated discourse.

The LDM is both a competence model of linguistic structure above the sentence level and a performance model. In the paper, we shall describe the linguistic discourse structuring resources and conventions available to speakers in carrying out communicative and interactional tasks and demonstrate how these resources are used in actual talk to create the complex discourses which speakers routinely produce and interpret. In our view, providing an adequate account of discourse structural relations is the first step towards what we believe to be the eventual goal of formal work in discourse understanding - the development of a system capable of assigning a proper semantic interpretation to every clause in a discourse.

L. Polanyi, "A Formal Syntax of Discourse," Technical Report of the Center for the Study of Reading, University of Illinois, 1986.

R. J. H Scha, B. C. Bruce and L. Polanyi, "Discourse Understanding," in Encyclopedia of Artificial Intelligence, S.C. Shapiro (ed.), John Wiley and Sons, New York, 1986 (also appears as Technical Report No 391, Center for the Study of Reading, University of Illinois, 1986).

Abstract

Research on natural language understanding has often focused on the problem of analyzing the structure and meaning of isolated sentences. To understand whole texts or dialogues, these sentences must be seen as elements whose significance resides in

the contribution they make to the larger whole. A computer natural language understanding system must interpret each sentence with respect to both the linguistic context, established by preceding sentences and the real-world setting. This paper reviews work on these issues, examining theories of the structure of discourse, the semantics of discourse, speech acts and pragmatics, and different communication modalities.

J. G. Schmolze, "Physics for Robots," Ph.D. Dissertation, University of Massachusetts, February 1986 (also a revised version to appear as BBN Technical Report No. 6222, Fall 1987).

C. L. Sidner, "Intentions, Attention and the Structure of Discourse," *Computational Linguistics*, Vol. 12, No. 3, 1986.

Abstract

In this paper we explore a new theory of discourse structure that stresses the role of purpose and processing in discourse. In this theory, discourse structure is composed of three separate but interrelated components: the structure of the sequence of utterances (called the linguistic structure), a structure of purposes (called the intentional structure), and the state of focus of attention (called the attentional state). The linguistic structure consists of segments of the discourse into which the utterances naturally aggregate. The intentional structure captures the discourse-relevant purposes, expressed in each of the linguistic segments as well as relationships among them. The attentional state is an abstraction of the focus of attention of the participants as the discourse unfolds. The attentional state, being dynamic, records the objects, properties, and relations that are salient at each point of the discourse. The distinction among these components is essential to provide an adequate explanation of such discourse phenomena as cue phrases, referring expressions, and interruptions.

The theory of attention, intention, and aggregation of utterances is illustrated in the paper with a number of example discourses. Various properties of discourse are described, and explanations for the behavior of cue phrases, referring expressions, and interruptions are explored

This theory provides a framework for describing the processing of utterances in a discourse. Discourse processing requires recognizing how the utterances of the discourse aggregate into segments, recognizing the intentions expressed in the discourse and the relationships among intentions, and tracking the discourse through the operation of the mechanisms associated with attentional state. This processing description specifies in these recognition tasks the role of information from the discourse and from the participants' knowledge of the domain.

N. S. Sridharan. "Semi-applicative Programming. Examples of Context-free Recognizers." BBN Report No. 6135, January 1986.

Abstract

Most current parallel programming languages are designed with a sequential programming language as the base language and have added constructs that allow parallel execution. We are experimenting with an applicative base langugage that has *implicit* parallelism everywhere, and then we introduce constructs that *inhibit* parallelism. The base language uses pure LISP as a foundation and blends in interesting features of Prolog and FP. Proper utilization of available machine resources is a crucial concern of programmers. We advocate several technqiues of controlling the behavior of functional programs without changing their meaning or functionality: program annotation with constructs that have benign side-effects, program transformation and adaptive scheduling. This conbination yeilds us a *semi-applicative* programming language and an interesting program methodology.

In this paper we deal with context-free parsing as an illustration of semi-applicative programming. Starting with the specification of a context-free recognizer, we have been successful in deriving variants of the recognition algorithm of Cocke-Kasami-Younger. One version is the CKY algorithm in parallel The second version includes a top-down predictor to limit the work done by the bottom-up recognizer. The third version uses a cost measure over derivations and produces minimal cost parses using a dynamic programming technique In another line of development, we arrive at a parallel version of the Earley algorithm. All of these algorithms reveal more concurrency than was apparent at first glance.

M. Vilain (with H. Kautz). "Constraint Propagation Algorithms for Temporal

Reasoning. *Proceedings of the Fifth Annual Conference on Artificial Intelligence (AAAI)*, Philadelphia, PA, pp. 377–382, August 1986.

Abstract

This paper considers computational aspects of several temporal representation languages. It investigates an interval-based representation, and a point-based one. Computing the consequences of temporal assertions is shown to be computationally intractable in the interval-based representation, but not in the point-based one. However, a fragment of the interval language can be expressed using the point language and benefits from the tractability of the latter.

9. PRESENTATIONS

B. Goodman, "Miscommunication and Plan Recognition," at the User Modelling Workshop, Maria Laach, West Germany, August 1986.

B. Goodman, "Reference and Reference Failures," Theoretical Issues in Natural Langauge Processing III (TINLAP3), New Mexico State University, Las Cruces, New Mexico, January 1987.

E. Hinrichs, "A Compositional Semantics for NP Reference and Aktionsarten", West Coast Conference on Formal Linguistics, March, 1986.

E. Hinrichs and L. Polanyi, "Pointing the Way: A Unified Treatment of Referential Gesture in Interactive Discourse," at the 22nd Annual Meeting of the Chicago Linguistic Society, April 17-19, 1986.

E. Hinrichs, "A Compositional Semantics for Directional Modifiers in English – Locative Case Reopened," at the 11th International Conference on Computational Linguistics, University of Bonn, August 25-29, 1986.

L. Polanyi, "Discourse Analysis from a Linguistic Point of View," Boston Interaction Research Group, January 29, 1986.

L. Polanyi, "A Linguistic Approach to Discourse Analysis," Massachusetts Interdisciplinary Discourse Analysis Seminar, February 5, 1986.

L. Polanyi, "A Formal Model of Discourse Structure," invited paper at 2nd Cognitive Science Seminar, Tel Aviv University, April 1-6, 1986.

L. Polanyi, "Discourse Syntax, Discourse Semantics, Discourse Semiotics: The Case of the Discourse Pivot," invited lecture, Cognitive Science Series, University of Buffalo, April 23, 1986.

L. Polanyi, "Narrative Organization and Disorganization," invited paper, Workshop Symposium on the Acquisition of Temporal Structures in Discourse, University of Chicago, April 16, 1986.

J. Schmolze, "Physics for Robots," at Brandeis University, March 1986.

J. Schmolze, "Physics for Robots," at AAAI-86, Philadelphia, August 11-15, 1986.

J. Schmolze, "Semantics for NIKL," MIT Workshop on Terminological Languages, Cambridge, MA, July 1986.

C. Sidner, "AI, Computational Linguistics and Discourse Theory, Massachusetts Interdisciplinary Discourse Analysis Seminar, March, 1986

C. Sidner, "Modelling Discourse Structure. The Role of Purpose in Discourse," invited talk, Sixth Annual Canadian AI Conference, Montreal, Canada, May 1986.

C. Sidner (with B. Grosz), "Plans in Discourse," SDF Benchmark Series in Computational Linguistics III, Plans and Intentions in Communication and Discourse, Monterey, California, March 1987.

N. S. Sridharan, "Semi-Applicative Programming. Examples of Context Free Recognizers," Technical Report No. 6135, BBN Laboratories Inc., January 1986.

N.S. Sridharan, Workshop on Future Directions in Computing, sponsored by the Army Research Office, Seabrook Island, SC, May 1986.

N.S. Sridharan, Workshop on Artificial Intelligence, sponsored by Ministry of Defense, India, in Bangalore, June 1986.

M.B. Vilain (with H. Kautz), "Constraint Propagation Algorithms for Temporal Reasoning," at AAAI-86, Philadelphia, August 11-15, 1986

M.B. Vilain, "Recent and Forthcoming Developments in KL-TWO," MIT Workshop on Terminological Languages, Cambridge, MA, July 1986.

Official Distribution List

Contract N00014-85-C-0079

	Copies
Scientific Officer Head, Information Sciences Division Office of Naval Research 800 North Quincy Street Arlington, VA 22217-5000 Attn: Dr. Alan L. Meyrowitz	1
Mr. Frank Skieber Defense Contract Administration Services Region - Boston 495 Summer Street Boston, MA 02210-2184	1
Director, Naval Research Laboratory Attn: Code 2627 Washington, DC 20375	1
Defense Technical Information Center Bldg. 5 Cameron Station Alexandria, VA 22314	12